



TUGAS AKHIR – TF 095565

**RANCANG BANGUN SISTEM PENGENDALIAN
FLOW RATE INLET KOLOM PURIFIKASI PADA
PROSES PURIFIKASI BIOGAS DENGAN WATER
SCRUBBER SYSTEM BERBASIS ATMEGA 128**

**RADIAN INDRA MUKROMIN
NRP 2414 031 034**

**Dosen Pembimbing
Hendra Cordova, ST, MT
NIP. 19690530 199412 1 001**

**PROGRAM STUDI D3 TEKNIK INSTRUMENTASI
DEPARTEMEN TEKNIK INSTRUMENTASI
FAKULTAS VOKASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2017**



TUGAS AKHIR – TF 095565

**RANCANG BANGUN SISTEM PENGENDALIAN
FLOW RATE INLET KOLOM PURIFIKASI PADA
PROSES PURIFIKASI BIOGAS DENGAN *WATER
SCRUBBER SYSTEM* BERBASIS ATMEGA 128**

RADIAN INDRA MUKROMIN
NRP 2414 031 034

Dosen Pembimbing
Hendra Cordova, ST, MT
NIP. 19690530 199412 1 001

PROGRAM STUDI D3 TEKNIK INSTRUMENTASI
DEPARTEMEN TEKNIK INSTRUMENTASI
FAKULTAS VOKASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2017



FINAL PROJECT - TF 095565

***DESIGN CONTROL SYSTEM FLOW RATE INLET
PURIFICATION COLUMN IN BIOGAS PURIFICATION
PROCESS WITH WATER SCRUBBER SYSTEM BASED
ATMEGA 128***

**RADIAN INDRA MUKROMIN
NRP 2414 031 034**

Supervisor
Hendra Cordova, ST, MT
NIP. 19690530 199412 1 001

***D3 INSTRUMENTATION ENGINEERING
DEPARTMENT OF INSTRUMENTATION ENGINEERING
FACULTY OF VOCATION
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2017***

**RANCANG BANGUN SISTEM PENGENDALIAN *FLOW*
RATE INLET KOLOM PURIFIKASI
PADA PROSES PURIFIKASI BIOGAS
DENGAN *WATER SCRUBBER SYSTEM* BERBASIS
ATMEGA 128**

TUGAS AKHIR

Oleh :

**Radian Indra Mukromin
NRP. 2414 031 034**

**Surabaya, 20 Juli 2017
Mengetahui / Menyetujui**

Dosen Pembimbing



**Hendra Cordova, ST, MT
NIP. 19690530 199412 1 001**



**Kepala Departemen
Teknik Instrumentasi**

**Dr.Ir. Purwadi Agus D,M.Sc
NIP. 19620822 198803 1 001**

**RANCANG BANGUN SISTEM PENGENDALIAN *FLOW*
RATE INLET KOLOM PURIFIKASI
PADA PROSES PURIFIKASI BIOGAS
DENGAN *WATER SCRUBBER SYSTEM* BERBASIS
ATMEGA 128**

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Ahli Madya
pada
Program Studi D3 Teknik Instrumentasi
Departemen Teknik Instrumentasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember

Oleh :

Radian Indra Mukromin

NRP. 2414 031 034

Disetujui oleh Tim Penguji Tugas Akhir :

1. Hendra Cordova, ST, MT^P (Dosen Pembimbing I)
2. Andi Rahmadiansah, ST, MT^S (Ketua Tim Penguji)
3. Ir. Tutug Dhanardono, MT^{Dhanardono} (Dosen Penguji II)
4. Arief Abdurrahman, ST, MT^{Arief} (Dosen Penguji III)

**SURABAYA
JULI 2017**

**RANCANG BANGUN SISTEM PENGENDALIAN *FLOW*
RATE INLET KOLOM PURIFIKASI
PADA PROSES PURIFIKASI BIOGAS
DENGAN WATER SCRUBBER SYSTEM BERBASIS
ATMEGA 128**

Nama Mahasiswa : RADIAN INDRA MUKROMIN
NRP : 2414 031 034
Jurusan : D III Teknik Instrumentasi,
Departemen Teknik Instrumentasi FV-ITS
Dosen Pembimbing : Hendra Cordova, ST, MT

ABSTRAK

Salah satu energy terbarukan yang terdapat di Indonesia adalah biogas, namun dalam penggunaannya saat ini masih banyak gas polutan yang terkandung di dalam produk biogas tersebut pada saat penggunaannya. Kondisi ini akibat dari tidak adanya sistem purifikasi yang ada di biogas. Pada teori Lantella, 2011, bahwa gas aliran dijaga konstan untuk menghasilkan isi gas produk Metana (CH₄) dan karbon dioksida (CO₂) sebesar 86-90% dan 4,5-8,0%. Pada perancangan sistem pengendalian flow ini, sensor yang digunakan untuk mengukur *flow* adalah sensor *flowmeter* 1". *Output* dari sensor *flowmeter* akan mengukur *flow* yang ada dan akan memberikan respon ke ATmega128 saat *setpoint flow* tercapai sehingga MOV sebagai aktuator *flow* akan merespon. Dari data respon pengendalian yang didapat menunjukkan bahwa pengendalian telah berjalan sesuai dengan *setpoint* yang diinginkan yaitu 5 liter/menit dan berosilasi mendekati setpoint. selain itu berdasarkan pengujian alat pada sistem pengendalian *flow* didapatkan *respon time* yang cukup baik untuk mencapai *stedy state* dengan maksimum *overshoot* 6 liter/menit. Waktu yang dibutuhkan untuk mencapai *set point* (*settling time*) dari awal pengambilan data adalah ± 25 detik.

Kata kunci : purifikasi, biogas, pendingin

**DESIGN CONTROL SYSTEM FLOW RATE INLET
PURIFICATION COLUMN IN BIOGAS PURIFICATION
PROCESS WITH WATER SCRUBBER SYSTEM
BASED ATMEGA 128**

Name of Student : RADIAN INDRA MUKROMIN
NRP : 2414 031 034
Departement : D III Teknik Instrumentasi,
Departemen Teknik Instrumentasi FV-ITS
Supervisor : Hendra Cordova, ST, MT

ABSTRACT

One of the renewable energy found in Indonesia is biogas, but in its current use there is still a lot of pollutant gas contained in the biogas product at the time of its use. This condition resulted from the absence of purification system in biogas. In Lantella's theory, 2011, that gas flow is kept constant To produce gas content of Methane (CH₄) and carbon dioxide (CO₂) products by 86-90% and 4.5-8.0%. In designing this flow control system, the sensor used to measure flow is a flowmeter sensor 1 ". The output of the flowmeter sensor will measure the current flow and will respond to ATmega128 when the setpoint flow is reached so that the MOV as actuator flow will respond. From the control response data obtained shows the desired changes in accordance with the desired setpoint of 5 liters / minute and oscillate close to the setpoint. In addition, based on testing at a good enough time to achieve stedy state with a maximum overshoot of 6 liters / minute. The time required to reach the set point (settling time) from the initial data retrieval is ± 25 seconds..

Keywords : Purification, Biogas, cooler

KATA PENGANTAR

Puji syukur kehadiran Allah SWT atas rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan laporan Tugas Akhir yang berjudul **“RANCANG BANGUN SISTEM PENGENDALIAN *FLOW RATE INLET* KOLOM PURIFIKASI PADA PROSES PURIFIKASI BIOGAS DENGAN *WATER SCRUBBER SYSTEM* BERBASIS ATMEGA 128”** dengan tepat waktu. terselesaikannya laporan ini juga tak luput dari dukungan dan peran serta dari orangtua dan keluarga besar serta berbagai pihak. Untuk itulah dalam kesempatan ini penulis mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Bapak Dr .Ir. Purwadi Agus D, M.Sc. selaku Ketua Departemen Teknik Instrumentasi FV-ITS.
2. Bapak Hendra Cordova, ST, MT dan Arief Abdurrahman ST., M.T selaku pembimbing Tugas Akhir yang telah membina dengan baik dan sabar.
3. Ibu Prof. Dr. Ir. Aulia Siti Aisjah, MT selaku Dosen Wali penulis.
4. Kedua orang tua dan kakak saya yang tidak henti-hentinya memberi semangat dan doa untuk menyelesaikan Tugas akhir ini
5. Bapak Hariyanto selaku pemilik biogas di desa Nongkojajar-Pasuruan-Jawa Timur yang memberikan biogas untuk pengambilan data penulis.
6. Seluruh Asisten Laboratorium *Microprocessor and Microcontroller* dan *Workshop Instrumentasi* , yang telah membantu dalam pengerjaan Tugas Akhir penulis.
7. *BIOGAS Team* yang telah membantu pengerjaan Tugas Akhir ini hingga selesai.
8. Teman-teman S1 Teknik Fisika dan D3 Teknik Instrumentasi angkatan 2014 FV-ITS.
9. Serta semua pihak yang tidak dapat disebutkan satu persatu.

Penulis menyadari bahwa laporan ini masih kurang sempurna. Oleh karena itu penulis menerima segala masukan baik berupa saran, kritik, dan segala bentuk tegur sapa demi kesempurnaan laporan ini.

Demikian laporan Tugas Akhir ini penulis persembahkan dengan harapan dapat bermanfaat dalam akademik baik bagi penulis sendiri maupun bagi pembaca.

Surabaya, 20 Juli 2016

Penulis.

DAFTAR ISI

	Hal
HALAMAN JUDUL.....	i
LEMBAR PENGESAHAN.....	iii
ABSTRAK.....	v
ABSTRACT	vi
KATA PENGANTAR	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xii

BAB I. PENDAHULUAN

1.1 Latar Belakang	1
1.2 Permasalahan.....	2
1.3 Tujuan	2
1.4 Batasan Masalah.....	3
1.5 Manfaat.....	3

BAB II. TINJAUAN PUSTAKA

2.1 Biogas	5
2.2 <i>Scrubber</i>	6
2.3 Proses Pemurnian Biogas	7
2.4 <i>System</i> Pengendalian Proses	8
2.5 <i>Flowmeter</i>	9
2.6 <i>Stepper Motor Controlled Valve</i>	10
2.7 Mikrokontroler ATmega.....	16
2.8 CodeVision AVR	16
2.9 Karakteristik Statik.....	17
2.10 Teori Ketidakpastian	19

BAB III. PERANCANGAN DAN PEMBUATAN ALAT

3.1 Diagram Alir (<i>Flowchart</i>)	21
3.2 Studi Literatur	22
3.3 Perancangan Sistem	23

3.4 Pembuatan Alat	27
3.5 Pengujian Alat dan Analisis	32

BAB IV. HASIL DATA DAN PEMBAHASAN

4.1 Rancang Bangun Alat	33
4.2 Pengujian Alat	33
4.3 Pengukuran Kalibrasi Sensor	36
4.4 Data karakteristik statik sensor	39
4.5 Pengendalian driver Motor Stepper pada MOV	42
4.6 Pembahasan	44

BAB V. PENUTUP

5.1 Kesimpulan.....	45
5.2 Saran.....	45

DAFTAR PUSTAKA

LAMPIRAN A (*Listing Program Pada Code Vision Avr*)

LAMPIRAN B (*Data Sheet System Atmega128*)

LAMPIRAN C (*Data Sheet Sensor Flow Type FS400A*)

LAMPIRAN D (*Manual Book*)

BIODATA PENULIS

DAFTAR GAMBAR

	Hal
Gambar 2.1 Prinsip Kerja <i>Dry Scrubber</i>	6
Gambar 2.2 Prinsip Kerja <i>Water Scrubber</i>	7
Gambar 2.3 <i>Flow control loop</i>	9
Gambar 2.4 <i>Flowmeter Turbin</i>	10
Gambar 2.5 <i>Stepper Motor Controlled Valve</i>	11
Gambar 2.6 Motor stepper dengan lilitan unipolar	13
Gambar 2.7 Motor stepper dengan lilitan bipolar	14
Gambar 2.8 <i>Mikrokontroler ATmega128</i>	16
Gambar 2.9 Tampilan CodeVision AVR	17
Gambar 2.10 Tabel T-student	20
Gambar 3.1 <i>Flowchart</i> Tugas Akhir	21
Gambar 3.2 Desain <i>Plant</i> Purifikasi Biogas	22
Gambar 3.3 P&id Purifikasi Biogas	22
Gambar 3.4 Diagram Blok Pengendalian Flow Inlet	23
Gambar 3.5 Diagram blok rangkaian driver motor stepper	24
Gambar 3.6 Simulasi Motor Stepper	24
Gambar 3.7 Rangkaian LCD 20x4	25
Gambar 3.8 Standar installasi pemasangan <i>flowmeter</i>	26
Gambar 3.9 Realisasi Pemasangan Sistem Kontrol <i>Flow</i>	29
Gambar 3.10 <i>Chip</i> pada <i>Code Vision AVR</i>	28
Gambar 3.11 <i>Buka Software</i>	28
Gambar 3.12 <i>Create New File</i>	29
Gambar 3.13 <i>Chip</i> pada <i>Code Vision AVR</i>	29
Gambar 3.14 <i>Alphanumeric LCD</i> pada <i>Code Vision AVR</i>	30
Gambar 3.15 konfigurasi pada PORTB	30
Gambar 3.16 Tombol <i>Generate</i>	31
Gambar 3.17 Tampilan Awal Program	31
Gambar 3.18 Tampilan <i>Listing</i> Program Sistem	32
Gambar 4.1 <i>Purification System Plant</i>	33
Gambar 4.2 Grafik Respon sistem	35
Gambar 4.3 Grafik Histerisis Sensor	41
Gambar 4.4 Motor stepper dua-phase (bipolar)	42
Gambar 4.5 Logika pengendalian motor stepper	43

DAFTAR TABEL

	Hal
Tabel 2.1 Pergerakan Half Step	15
Tabel 2.2 Pergerakan Full Step	15
Tabel 4.1 Hasil Pengujian Sistem	34
Tabel 4.2 Hasil Pengambilan Data Naik.....	36
Tabel 4.3 Hasil Pengambilan Data Turun.....	36
Tabel 4.4 Data Kalibrasi Sensor a	37
Tabel 4.5 Data Kalibrasi Sensor b	37
Tabel 4.6 Hasil Data Sensor Naik dan Turun	40
Tabel 4.7 Tabel kebenaran pengendalian motor stepper	43

BAB I

PENDAHULUAN

1.1 Latar Belakang

Ketergantungan terhadap energy berbasis fosil dialami hampir disetiap Negara bukan hanya Indonesia ketergantungan energi fosil masih didominasi oleh kebutuhan minyak yang mencapai 41,8[1]. Saat ini cadangan energi fosil sudah semakin menipis dan pencarian sudah sangat sulit dilakukan, Sedangkan pertumbuhan konsumsi energy Indonesia lebih tinggi dibandingkan pertumbuhan konsumsi dunia. Untuk mengurangi ketergantungan terhadap energi fosil tersebut Kementerian Energi dan Sumber Daya Mineral Republik Indonesia merencanakan pada tahun 2025 peran energi terbarukan mencapai 17% dari total bauran energi primer. Salah satu energy terbarukan yang terdapat di Indonesia adalah biogas dengan skala rumah tangga, namun dalam penggunaannya saat ini masih banyak gas polutan yang terkandung di dalam produk biogas tersebut pada saat penggunaannya.

Biogas dihasilkan dari proses anaerobik dan terdiri dari beberapa unsur. Komposisi biogas yang dihasilkan dari pencernaan anaerobik kurang-lebih sekitar 60 - 70 % CH₄, sekitar 30 - 40% CO₂, kurang dari 1% N₂, dan sekitar 10 - 2000 ppm H₂S [2]. Gas metana menjadi unsur primer dan menjadi parameter utama dalam menentukan kualitas sebuah produk biogas.

Biogas telah banyak dimanfaatkan oleh masyarakat Indonesia khususnya pada daerah peternakan sapi karena dapat menjadi sumber alternative. Provinsi Jawa Timur merupakan provinsi dengan jumlah reaktor BIRU terbanyak secara nasional .Tercatat sebanyak 1.350 unit reaktor biogas yang tersebar di 12 desa di Kecamatan Tukur. Angka tersebut sama dengan menampung kurang lebih 1.450 Kepala Keluarga (Hariyanto, 2015). daerah tersebut komposisi biogasnya masih mengandung gas polutan berupa CO₂ sebesar 26,93% dan H₂S sebesar 41,76

ppm[3]. Kondisi ini akibat dari tidak adanya sistem purifikasi yang ada di biogas, sehingga dapat berdampak pada terancamnya kesehatan masyarakat pengguna biogas. penelitian Hermanto, 2016, bahwa pengaruh konsentrasi NaOH dan laju aliran gas pada proses pemurnian biogas perlakuan laju aliran gas dijaga konstan pada 3,75 mL / min. untuk menurunkan kandungan CO₂ sebesar 71,9% kandungan H₂S sebesar 100%.

Dari beberapa tinjauan tersebut, maka dalam Tugas Akhir ini dirancang sistem pengendalian *flow inlet* gas pada purifikasi biogas menggunakan water scrubber dengan tujuan meningkatkan efektifitas reduksi H₂S. Hal ini didasarkan pada teori Lantella, 2011, bahwa gas aliran dijaga konstan pada 7,41 Nm³ / h. untuk menghasilkan isi gas produk Metana (CH₄) dan karbon dioksida (CO₂) sebesar 86-90% dan 4,5-8,0%

1.2 Perumusan Masalah

Berdasarkan latar belakang yang dijelaskan diatas, maka rumusan masalah dalam Tugas Akhir mengenai pengendalian *Flow* pada *inlet* purifikasi biogas ini adalah sebagai berikut :

- Bagaimana membangun sebuah rancang bangun sistem pengendalian *Flow inlet* pada proses purifikasi biogas ?
- Bagaimana nilai karakteristik static dan hasil uji dari respon sistem ?

1.3 Tujuan Tugas Akhir

Tujuan utama dari penelitian ini adalah untuk memenuhi mata kuliah tugas akhir sebagai syarat kelulusan dari program studi diploma 3 metrologi dan instrumentasi serta hasil dari penelitian tersebut diharapkan dapat digunakan sebagai energi alternatif di masa mendatang dan untuk memberikan solusi pada rumusan masalah diatas adalah sebagai berikut :

- Untuk membangun sebuah rancang bangun sistem pengendalian *Flow* pada *inlet* purifikasi biogas.
- Untuk mengetahui nilai karakteristik static dan hasil uji dari respon sistem.

1.4 Batasan Masalah

Perlu diberikan beberapa batasan permasalahan agar pembahasan tidak meluas dan menyimpang dari tujuan. Adapun batasan permasalahan dari sistem yang dirancang ini yaitu :

- Variabel yang dikendalikan adalah *flow*.
- Sensor yang digunakan adalah *Flowmeter G 1"*.
- Menggunakan Microcontroller Atmega 128 sebagai fungsi kontroler.
- Aktuator yang digunakan adalah *motor stepper*.
- Biogas yang digunakan berasal dari kotoran sapi.
- Uji coba menggunakan ban penampung yang berisi biogas.

1.5 Manfaat

Manfaat yang didapatkan dalam mengerjakan tugas akhir ini mahasiswa atau penulis paham dan mengerti dalam merancang dan membangun suatu sistem mengenai pengendalian *Flow* pada *inlet* purifikasi biogas.

(Halaman ini sengaja dikosongkan)

BAB II

TINJAUAN PUSTAKA

2.1 Biogas

Biogas mulai berkembang sejak abad 17 di Perancis. Biogas sendiri merupakan gas yang dihasilkan dari proses penguraian bahan-bahan organik dalam kondisi anaerobik atau minim oksigen. Bahan-bahan organik tersebut diantaranya bisa berupa daun-daun yang telah membusuk, kotoran manusia, kotoran ternak, dan masih banyak lagi selama masih tergolong bahan organik. Di Indonesia bahan organik untuk biogas yang paling banyak digunakan adalah dari kotoran hewan karena masih banyaknya peternakan. Sehingga potensi peternakan di Indonesia selain untuk kebutuhan pangan, juga bisa untuk pengembangan sumber daya terbarukan seperti biogas.

Biogas merupakan gas yang sangat mudah terbakar. Unsur-unsur utama biogas diantaranya metana (CH_4), karbon dioksida (CO_2), beberapa unsur lain seperti amonia (NH_3), hidrogen sulfida (H_2S), karbon monoksida (CO), hidrogen (H_2), nitrogen (N_2), dan oksigen (O_2). Komposisi biogas yang dihasilkan dari pencernaan anaerobik biasanya sekitar 60 - 70 % CH_4 , sekitar 30 - 40% CO_2 , kurang dari 1% N_2 , dan sekitar 10 - 2000 ppm H_2S [2]. Biogas mudah terbakar karena kandungan gas metana didalamnya. Gas metana memiliki nilai kalor yang tinggi. Untuk gas metan murni (100 %) mempunyai nilai kalor 8900 Kkal/ m^3 . Ketika dibakar 1 ft^3 gas bio menghasilkan sekitar 10 BTU (2,52 Kkal) energi panas per persentase komposisi metan [2].

Biogas memiliki berbagai keunggulan dibanding dengan bahan bakar lain. Biogas lebih ramah lingkungan karena tidak seperti bahan bakar dari fosil yang jika dilakukan pembakaran menghasilkan gas CO_2 . Sampah-sampah organik yang dibiarkan membusuk bisa menjadi gas metana yang berbahaya jika langsung lepas ke udara, sehingga dengan pemanfaatannya sebagai biogas bisa menekan polusi udara yang ditimbulkan oleh gas metana. Selain itu, dengan memanfaatkan kotoran hewan peternakan sebagai biogas, otomatis dapat menambah penghasilan

para peternak atau bisa mensejahterakan hidup para peternak dengan memanfaatkan biogas tersebut sendiri.

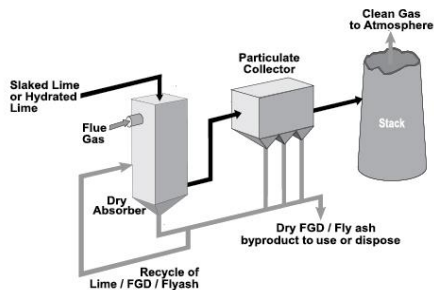
2.2 Scrubber

scrubber merupakan alat pemisahan suatu partikel solid (debu) yang ada di gas atau udara dengan menggunakan cairan sebagai alat bantu. Air adalah cairan yang pada umumnya digunakan dalam proses *scrubbing*, meskipun dapat juga digunakan cairan lainnya seperti asam sulfide.

Terdapat 2 jenis *scrubber* yaitu *Dry Scrubber* dan *Wet Scrubber*. Perbedaan dari kedua jenis *scrubber* tersebut yaitu penggunaan fluida cair serta pengendalian temperaturnya.

a. Dry Scrubber

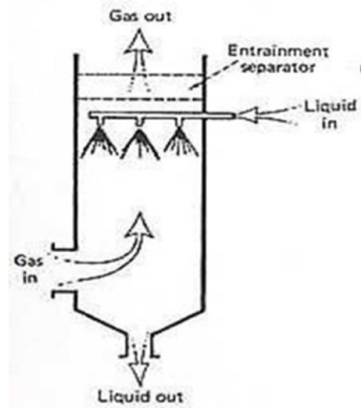
Dry Scrubber merupakan pengontrol polusi yang berlangsung dalam proses kering sehingga penggunaannya lebih banyak untuk partikel padat. Hal ini juga disebabkan karena dalam proses kerja *Dry Scrubber* hanya bisa memproses emisi padat seperti *fly ash* (partikel padat). Proses kerja dari *Dry Scrubber* tidak banyak berpengaruh terhadap pengendalian gas karena hanya fokus dalam pengendalian partikel padat. Prinsip kerjanya yaitu dengan mengendalikan aliran gas yang mengandung partikel padat.



Gambar 2.1 Prinsip Kerja *Dry Scrubber*

b. *Water Scrubber*

Sesuai dengan namanya, *Water Scrubber* menggunakan fluida cair pada proses kerjanya. Kebanyakan fluida yang digunakan adalah air. Hal inilah yang menjadi perbedaan antara *Dry Scrubber* dengan *Wet Scrubber*. Penggunaan fluida tersebut memberi kelebihan pada sistem *Water Scrubber* karena dapat mengontrol gas yang dapat larut seperti Sox, NOx, H₂S, yang pada *Dry Scrubber* polutan tersebut tidak dapat dikontrol dengan baik. Kelebihan yang lain yaitu kontrol temperatur karena pada sistem *Water Scrubber* terjadi penggabungan 2 jenis fluida yang berbeda temperaturnya. Untuk cara kerja *Water Scrubber* lebih efisien dibandingkan *Dry Scrubber* karena mampu menangkap partikel dalam bentuk kecil serta dapat mengikat gas. Cara kerjanya yaitu mengalirkan fluida cair pada aliran gas sehingga langsung difiltrasi oleh fluida cair tersebut.



Gambar 2.2 Prinsip Kerja *Water Scrubber*[4]

2.3 Proses Pemurnian Biogas

Unsur-unsur utama pada biogas diantaranya metana (CH₄), karbon dioksida (CO₂), beberapa unsur lain seperti amonia (NH₃), hidrogen sulfida (H₂S), karbon monoksida (CO), hidrogen (H₂), nitrogen (N₂), dan oksigen (O₂). Suatu biogas dapat dilihat

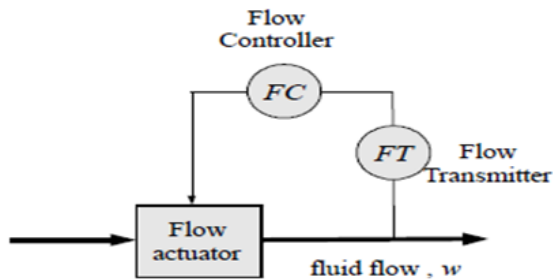
kualitasnya dari banyaknya gas metana yang dikandung, karena semakin banyak kandungan metana, maka energi yang dihasilkan juga semakin besar. Namun adanya gas pengotor yang berupa air, CO_2 , dan H_2S membuat kualitas dari biogas tersebut tidak maksimal. Peningkatan kualitas biogas bisa dilakukan dengan menghilangkan gas pengotor tersebut terutama CO_2 dan H_2S . Hal ini disebabkan karena hidrogen sulfur mengandung racun dan zat yang menyebabkan korosi. Jika biogas mengandung senyawa ini, maka akan menimbulkan gas yang berbahaya. Jika biogas dibakar, H_2S akan membentuk senyawa baru berupa bersama oksigen berupa sulfur dioksida (SO_2) atau sulfur trioksida (SO_3). Kedua senyawa ini lebih beracun darip H_2S . Dan jika terus dilanjutkan, maka akan terbentuk sulfur acid (H_2SO_3) yang bersifat lebih korosif daripada H_2S .

Selain itu, pembakaran gas yang masih mengandung H_2S juga akan menghasilkan senyawa asam (H_2SO_4) yang bisa menimbulkan hujan asam. Hal ini dapat menimbulkan kerugian besar karena bersifat korosif dan mencemari lingkungan hidup. Sedangkan CO_2 memiliki sifat yang dapat menghambat proses pembakaran yang sempurna, beracun, dan dapat menyebabkan korosi.

2.4 System Pengendalian Proses

Hampir semua proses di industri membutuhkan peralatan – peralatan otomatis untuk mengendalikan parameter – parameter prosesnya. Otomatisasi tidak hanya diperlukan demi kelancaran operasi, keamanan, ekonomi maupun mutu produk, tetapi lebih merupakan kebutuhan pokok. Suatu proses industri tidak akan dapat dijalankan tanpa bantuan sistem pengendalian. Parameter – parameter yang umum dikendalikan dalam suatu proses adalah tekanan (*pressure*), laju aliran (*flow*), suhu (*temperature*) dan permukaan zat cair (*level*). Gabungan serta alat – alat pengendalian otomatis itulah yang disebut sistem pengendalian proses (*process control system*). Sedangkan semua peralatan yang membentuk sistem pengendalian tersebut disebut instrumentasi pengendalian proses (*process control instrumentation*).

Secara garis besar suatu rangkaian pengendalian proses dibagi menjadi 4 langkah, yaitu mengukur – membandingkan – menghitung – mengkoreksi. Langkah pertama yaitu mengukur, mengukur merupakan tugas dari sensor. Langkah berikutnya adalah membandingkan apakah hasil pengukuran dari sensor sudah sesuai dengan apa yang dikehendaki. Apabila terjadi ketidaksesuaian antara *set point* dengan hasil pengukuran maka akan dilakukan perhitungan untuk menentukan aksi apa yang dilakukan supaya sesuai dengan set point yang diinginkan. Pada langkah kedua dan ketiga ini adalah tugas dari pengendali. Langkah terakhir adalah melakukan pengkoreksian yang merupakan tugas dari aktuatur.



Gambar 2.3 *Flow control loop*[5]

2.5 Flowmeter

Flowmeters turbin menggunakan energi mekanik dari cairan yang mengalir guna memutar sebuah “pinwheel” (rotor). Blades pada rotor dibuat miring dengan sudut tertentu layaknya baling-baling, untuk mengubah energi dari aliran fluida menjadi energi rotasi. Poros rotor berputar pada bushing/bearing dimana ketika liquid bergerak lebih cepat pada flowmeter turbin, maka akan lebih banyak menghasilkan pulsa yang akan sebanding dengan kecepatan aliran fluida. Sensor pick-up pada flowmeter turbin akan memproses sinyal pulsa untuk menentukan aliran fluida.



Gambar 2.4 *Flowmeter Turbin*

Dimana output dari *water flowmeter* ini yaitu frekuensi. Persamaan untuk merubah frekuensi ke debit yaitu :

$$F(\text{Hz}) = 4.8 \times Q (\text{L/min})$$

Spesifikasi G 1 inch *water flow sensor* :

- Rentang tegangan operasional antara 5 ~ 24 Volt DC
- Maksimum arus operasional sebesar 15 mA (pada 5 VDC)
- Maksimum suhu air 120°C
- Maksimum suhu operasional 80°C
- Rentang kelembapan operasional antara 35% ~ 90% RH
- Maksimum tekanan air 1,2 MPa
- Diameter *thread inlet/outlet* 1" (~25 mm)

2.6 *Stepper Motor Controlled Valve*

Salah satu jenis *motor control valve* dc yang dikendalikan dengan pulsa-pulsa digital. *Prinsip* kerja motor stepper adalah bekerja dengan mengubah pulsa elektronis menjadi gerakan mekanis diskrit dimana motor stepper bergerak berdasarkan urutan pulsa yang diberikan kepada motor stepper tersebut.



Gambar 2.5 *Stepper Motor Controlled Valve*

Stepping motor memiliki beberapa keunggulan-keunggulan sebagai berikut:

- a. Mudah dikendalikan dengan putaran sinkron yang hanya tergantung pada denyut yang diterima.
- b. Memiliki sudut yang tepat pada setiap langkah tanpa adanya akumulasi kesalahan sudut.
- c. Memiliki respon memulai dan berhenti yang baik pada saat digunakan, dan dapat digunakan sebagai servo motor.
- d. Putaran sinkron pada kecepatan yang sangat rendah dapat dicapai selagi “*carriage*” dihubungkan langsung pada shaft *stepping motor*.
- e. Memiliki ketahanan yang tinggi karena masa pengoperasiannya tergantung pada ketahanan “*bearing*” dan tidak pada bagian yang menempel semisal sikat atau “*brush*”.
- f. Berbagai macam sudut dapat dicapai tergantung dari tipe motor.
- g. Memiliki torsi menahan di posisi yang diinginkan pada saat motor dalam keadaan diam.

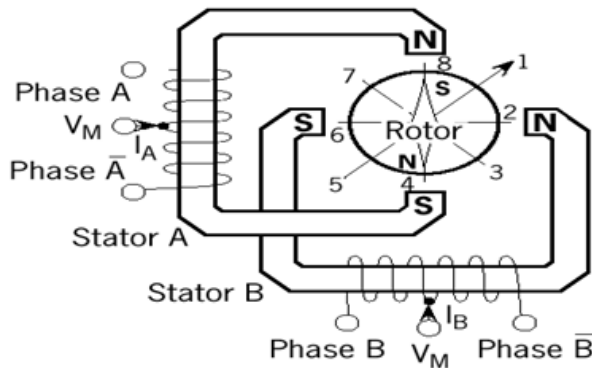
- h. Tidak terdapat sisa arus yang mengalir meskipun putaran terhalang oleh kelebihan beban.
- i. Kecepatan putaran tidak dipengaruhi oleh pergantian beban, karenanya kecepatan dapat selalu terjaga secara simultan dengan denyut masuk.

Motor stepper disini berfungsi sebagai penggerak valve secara elektrik. Motor stepper adalah perangkat elektromekanis yang bekerja dengan mengubah pulsa elektronis menjadi gerakan mekanis diskrit. Motor stepper bergerak berdasarkan urutan pulsa yang diberikan kepada motor. Karena itu, untuk menggerakkan motor stepper diperlukan pengendali motor stepper yang membangkitkan pulsa-pulsa periodik. Penggunaan motor stepper memiliki beberapa keunggulan dibandingkan dengan penggunaan motor DC biasa. Keunggulannya antara lain adalah :

- Sudut rotasi motor proporsional dengan pulsa masukan sehingga lebih mudah diatur.
- Motor dapat langsung memberikan torsi penuh pada saat mulai bergerak
- Posisi dan pergerakan repetisinya dapat ditentukan secara presisi
- Memiliki respon yang sangat baik terhadap mulai, stop dan berbalik (perputaran)
- Sangat realibel karena tidak adanya sikat yang bersentuhan dengan rotor seperti pada motor DC
- Dapat menghasilkan perputaran yang lambat sehingga beban dapat dikopel langsung ke porosnya
- Frekuensi perputaran dapat ditentukan secara bebas dan mudah pada range yang luas.

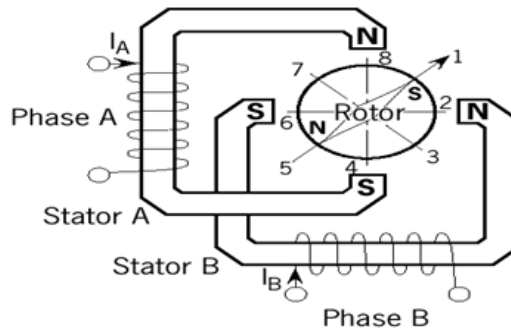
Berdasarkan metode perancangan rangkain pengendalinya, motor stepper dapat dibagi menjadi jenis unipolar dan bipolar. Rangkaian pengendali motor stepper unipolar lebih mudah dirancang karena hanya memerlukan satu switch / transistor setiap lilitannya. Untuk menjalankan dan menghentikan motor ini cukup dengan menerapkan pulsa digital yang hanya terdiri atas

tegangan positif dan nol (ground) pada salah satu terminal lilitan (wound) motor sementara terminal lainnya dicatu dengan tegangan positif konstan (V_M) pada bagian tengah (center tap) dari lilitan.



Gambar 2.6 Motor stepper dengan lilitan unipolar[6]

Untuk motor stepper dengan lilitan bipolar, diperlukan sinyal pulsa yang berubah-ubah dari positif ke negatif dan sebaliknya. Jadi pada setiap terminal lilitan (A & B) harus dihubungkan dengan sinyal yang mengayun dari positif ke negatif dan sebaliknya. Karena itu dibutuhkan rangkaian pengendali yang agak lebih kompleks daripada rangkaian pengendali untuk motor unipolar. Motor stepper bipolar memiliki keunggulan dibandingkan dengan motor stepper unipolar dalam hal torsi yang lebih besar untuk ukuran yang sama.



Gambar 2.7 Motor stepper dengan lilitan bipolar[6]

Motor stepper bergerak per langkah, dimana setiap langkah mempunyai derajat pergerakan yang sama tergantung dari resolusi dari motor tersebut. Motor stepper yang mempunyai resolusi pergerakan yang kecil, pergerakannya lebih baik dibandingkan resolusi yang besar. Hal ini disebabkan karena pergerakan yang besar akan menghasilkan gerakan yang lebih kasar dibandingkan dengan motor yang memiliki resolusi kecil. Motor stepper dapat dikendalikan secara full step dan half step. Pengendalian secara half step lebih baik daripada pengendalian secara full step karena dengan pengendalian half step pergerakan dari motor lebih halus daripada menggunakan pengendalian dengan pengendalian full step.

Half Step adalah cara mengendalikan motor stepper sehingga menghasilkan pergerakan motor yang lebih halus. Karena pergerakan rotor dalam motor stepper yang bergerak dengan sudut sebesar $\frac{1}{2}$ derajat dari besar sudut antara 2 buah kutub (coil) yang berdekatan, sehingga pergerakan yang dihasilkan lebih halus. [7]

Tabel 2.1 Pergerakan Half Step [7]

Step	Coil 4	Coil 3	Coil 2	Coil 1	Pergerakan Motor
1	1	0	0	0	
2	1	1	0	0	
3	0	1	0	0	
4	0	1	1	0	

Full step adalah cara mengendalikan motor stepper sehingga dihasilkan pergerakan motor namun tidak sehalus pergerakan half step. Hal ini disebabkan karena pergerakan rotor dalam motor stepper yang bergerak per 1 buah kutub (coil). Untuk lebih jelas, konfigurasi motor stepper dengan pengendalian full step untuk setiap pergerakan motor adalah sebagai berikut :

Tabel 2.2 Pergerakan Full Step [7]

Step	Coil 4	Coil 3	Coil 2	Coil 1	Pergerakan Motor
1	1	0	0	0	
2	0	1	0	0	
3	0	0	1	0	
4	0	0	0	1	

2.7 Mikrokontroler ATmega

Mikrokontroler merupakan keseluruhan sistem komputer yang dikemas menjadi sebuah *chip* di mana di dalamnya sudah terdapat Mikroprosesor, I/O, Memori bahkan ADC, berbeda dengan Mikroprosesor yang berfungsi sebagai pemroses data (Heryanto, dkk, 2008:1).

Mikrokontroler AVR (*Alf and Vegard's Risc processor*) memiliki arsitektur 8 bit, dimana semua instruksi dikemas dalam kode *16-bit* dan sebagian besar instruksi dieksekusi dalam 1 siklus *clock* atau dikenal dengan teknologi RISC (*Reduced Instruction Set Computing*). Secara umum, AVR dapat dikelompokkan ke dalam 4 kelas, yaitu keluarga AT90Sxx, keluarga ATmega dan AT86RFxx. Pada dasarnya yang membedakan masing-masing adalah kapasitas memori, *peripheral* dan fungsinya (Heryanto, dkk, 2008:1). Dari segi arsitektur dan instruksi yang digunakan, mereka bisa dikatakan hampir sama. Berikut ini gambar Mikrokontroler Atmega128.



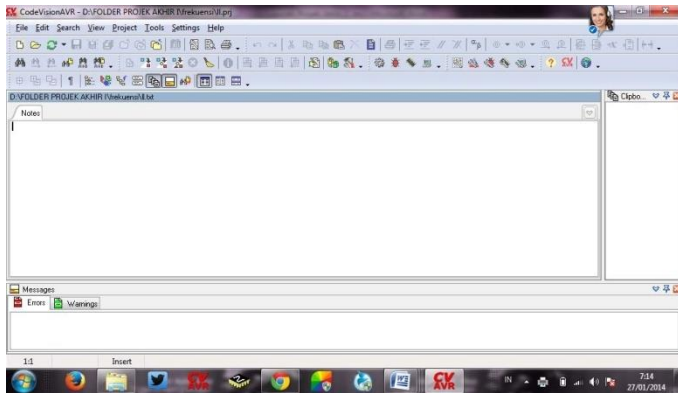
Gambar 2.8 Mikrokontroler ATmega128

2.8 CodeVision AVR

CodeVision AVR merupakan sebuah *software* yang digunakan untuk memprogram mikrokontroler sekarang ini telah umum. Mulai dari penggunaan untuk kontrol sederhana sampai kontrol yang cukup kompleks, mikrokontroler dapat berfungsi

jika telah diisi sebuah program, pengisian program ini dapat dilakukan menggunakan compiler yang selanjutnya diprogram ke dalam mikrokontroler menggunakan fasilitas yang sudah disediakan oleh program tersebut. Salah satu *compiler* program yang umum digunakan sekarang ini adalah CodeVision AVR yang menggunakan bahasa pemrograman C.

CodeVision AVR mempunyai suatu keunggulan dari compiler lain, yaitu adanya *codewizard*, fasilitas ini memudahkan pengguna atau pemrogram dalam inisialisasi mikrokontroler yang akan digunakan^[13].



Gambar 2.9 Tampilan CodeVision AVR

2.8 Karakteristik Statik

Karakteristik statik pengukuran merupakan karakteristik yang ditentukan melalui perhitungan matematik atau secara grafik. Karakteristik statik merupakan karakter yang menggambarkan parameter dari sebuah instrument pada saat keadaan *steady*. [5] Karakteristik statik terdiri dari:

a. *Range*

Range merupakan selisih antara nilai minimum dan maksimum yang terukur oleh suatu instrument atau alat ukur.

b. *Akurasi*

Akurasi merupakan tingkat ketelitian suatu alat dalam memberikan hasil pengukuran.

c. Toleransi

Toleransi menunjukkan kesalahan maksimum yang diperbolehkan pada hasil pembacaan alat ukur.

d. Sensitivitas

Sensitivitas menunjukkan perubahan output instrument yang terjadi saat diberi kualitas pengukuran yang berbeda. Sensitivitas diperoleh dari $\Delta O / \Delta I$, dimana O adalah output dan I adalah input.

e. Repeatability

Repeatability merupakan kemampuan *instrument* dalam menampilkan ulang output pengukuran yang sama pada pengukuran yang berulang.

f. Linearitas

Linearitas merupakan grafik yang menampilkan pengukuran nilai sebenarnya (input) yang dapat menghasilkan output nilai yang ditunjukkan oleh instrument (output). Pengukuran yang baik adalah ketika input pengukuran dan output pengukuran berbanding lurus (linear). Linearitas dapat diperoleh dari persamaan berikut:

$$O - O_{\min} = \frac{O_{\max} - O_{\min}}{I_{\max} - I_{\min}} (I - I_{\min}) \quad 2.1$$

Persamaan Linearitas :

$$O_{\text{ideal}} = KI + a \quad 2.2$$

Dimana:

$$K = \frac{O_{\max} - O_{\min}}{I_{\max} - I_{\min}} \quad 2.3$$

Dan

$$a = O_{\min} - KI_{\min} \quad 2.4$$

dimana :

$$O = 1.6 \times 10^{-3}I + 4.0 \quad 2.5$$

2.9 Teori Ketidakpastian

Ketidakpastian pengukuran merupakan tingkat seberapa besar ketidakpastian yang dihasilkan oleh suatu alat ukur. Dalam menghitung ketidakpastian pengukuran ada beberapa langkah yang harus dihitung, antara lain:

a. Koreksi

Koreksi dapat diperoleh dengan persamaan berikut

Koreksi = Pembacaan standard-Pembacaan alat

b. Standard deviasi

$$\sigma = \frac{\sqrt{\sum (D_i - D')^2}}{n-1} \quad 2.6$$

dimana :

D_i = koreksi alat ukur

D' = rata-rata koreksi

n = Banyak range pengukuran

c. Analisa Type A, (U_a)

Pada analisa tipe A ini hasilnya diperoleh dari data pengukuran. Adapun persamaannya adalah sebagai berikut:

$$U_{a1} = \frac{\sigma_{maks}}{\sqrt{n}} \quad 2.7$$

$$U_{a2} = \sqrt{\frac{SSR}{n-2}} \quad 2.8$$

d. Analisa Type B, (U_b)

Analisa tipe B ini diperoleh berdasarkan sertifikat kalibrasi atau spesifikasi dari alat ukur. Adapun persamaannya adalah sebagai berikut:

$$U_{b1} = \frac{Resolusi/2}{\sqrt{3}} \quad 2.9$$

Dimana :

SSR = Sum Square Residual

$$U_{b2} = \frac{a}{k} \quad 2.10$$

e. Ketidakpastian Kombinasi (UC)

Uc merupakan Ketidakpastian kombinasi dari ketidakpastian tipe A dan ketidakpastian tipe B. Adapun persamaan dari ketidakpastian kombinasi adalah:

$$UC = \sqrt{Ua1^2 + Ua2^2 + Ub1^2 + Ub2^2} \quad 2.11$$

f. Ketidakpastian Diperluas

Hasil akhir kalibrasi adalah ketidakpastian diperluas sehingga alat ukur tersebut dapat diketahui ketidakpastiannya melalui U_{expand} . Persamaan U_{expand} adalah:

$$U_{expand} = k \cdot UC \quad [6] \quad 2.12$$

Untuk mencari nilai k, maka melihat table t student sesuai dengan *confidence level* 95%. Tabel T *student* dapat dilihat pada gambar 2.1

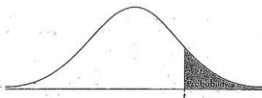


TABLE B: t-DISTRIBUTION CRITICAL VALUES

	Tail probability p											
df	.35	.30	.25	.20	.15	.10	.05	.025	.02	.01	.005	.0025
1	1.000	1.376	1.963	3.078	6.314	12.71	15.89	31.82	63.66	127.3	318.3	636.6
2	1.161	1.386	1.886	2.924	4.303	6.981	10.88	17.00	24.46	31.59	42.91	63.66
3	1.250	1.393	1.893	2.877	4.177	6.581	10.24	16.01	23.68	30.82	41.78	62.56
4	1.345	1.401	1.901	2.839	4.045	6.256	9.600	15.33	22.32	29.30	40.33	61.26
5	1.457	1.408	1.910	2.801	3.919	5.965	8.946	14.55	21.48	28.38	39.43	60.00
6	1.571	1.415	1.918	2.764	3.793	5.688	8.327	13.75	20.70	27.48	38.49	58.88
7	1.697	1.422	1.926	2.728	3.669	5.408	7.709	12.96	19.95	26.67	37.46	57.78
8	1.833	1.429	1.934	2.693	3.540	5.133	7.093	12.17	19.25	25.85	36.43	56.69
9	1.978	1.436	1.941	2.658	3.413	4.853	6.479	11.39	18.55	25.01	35.40	55.61
10	2.132	1.443	1.948	2.625	3.287	4.571	5.861	10.61	17.86	24.15	34.38	54.54
11	2.292	1.450	1.955	2.592	3.161	4.288	5.241	9.850	17.16	23.28	33.36	53.48
12	2.457	1.457	1.962	2.560	3.036	4.003	4.621	9.089	16.47	22.39	32.35	52.43
13	2.637	1.464	1.969	2.528	2.911	3.715	4.257	8.332	15.78	21.49	31.34	51.39
14	2.821	1.471	1.976	2.497	2.786	3.426	3.900	7.581	15.09	20.60	30.33	50.36
15	3.009	1.478	1.983	2.467	2.661	3.137	3.581	6.848	14.40	19.71	29.32	49.34
16	3.201	1.485	1.990	2.437	2.536	2.849	3.257	6.118	13.71	18.83	28.31	48.33
17	3.397	1.492	1.997	2.407	2.411	2.567	2.932	5.391	13.02	17.95	27.30	47.33
18	3.597	1.499	2.004	2.377	2.286	2.276	2.607	4.668	12.33	17.09	26.29	46.34
19	3.801	1.506	2.011	2.348	2.161	2.000	2.286	3.947	11.64	16.23	25.28	45.36
20	4.009	1.513	2.018	2.319	2.036	1.724	2.000	3.226	10.95	15.38	24.27	44.39
21	4.221	1.520	2.025	2.290	1.911	1.448	1.724	2.506	10.26	14.53	23.26	43.43
22	4.437	1.527	2.032	2.261	1.786	1.172	1.448	1.986	9.57	13.68	22.25	42.48
23	4.657	1.534	2.039	2.232	1.661	0.896	1.172	1.748	8.88	12.83	21.24	41.54
24	4.881	1.541	2.046	2.203	1.536	0.620	0.896	1.510	8.19	11.98	20.23	40.61
25	5.108	1.548	2.053	2.174	1.411	0.344	0.620	1.272	7.50	11.13	19.22	39.69
26	5.339	1.555	2.060	2.145	1.286	0.068	0.344	1.034	6.81	10.28	18.21	38.78
27	5.573	1.562	2.067	2.116	1.161	-0.208	0.068	0.796	6.12	9.43	17.20	37.88
28	5.811	1.569	2.074	2.087	1.036	-0.482	-0.208	0.558	5.43	8.58	16.19	36.99
29	6.053	1.576	2.081	2.058	0.911	-0.756	-0.482	0.320	4.74	7.73	15.18	36.11
30	6.299	1.583	2.088	2.029	0.786	-1.030	-0.756	0.082	4.05	6.88	14.17	35.24
40	7.153	1.614	2.120	1.881	0.259	-1.771	-1.030	-0.043	2.70	5.19	12.92	32.85
50	7.962	1.641	2.150	1.733	-0.259	-2.009	-1.771	-0.109	1.93	4.53	11.75	30.47
60	8.591	1.665	2.176	1.585	-0.756	-2.209	-2.009	-0.175	1.39	3.98	10.59	28.13
70	9.144	1.688	2.200	1.437	-1.256	-2.377	-2.209	-0.241	0.94	3.53	9.45	25.82
80	9.600	1.709	2.221	1.289	-1.756	-2.519	-2.377	-0.307	0.60	3.18	8.33	23.52
90	9.984	1.728	2.240	1.141	-2.256	-2.639	-2.519	-0.373	0.36	2.92	7.29	21.29
100	10.295	1.745	2.257	1.000	-2.756	-2.744	-2.639	-0.439	0.21	2.71	6.31	19.15
∞	6.74	841	1.036	1.282	1.645	1.960	2.054	2.326	2.576	2.807	3.091	3.291

Confidence level C

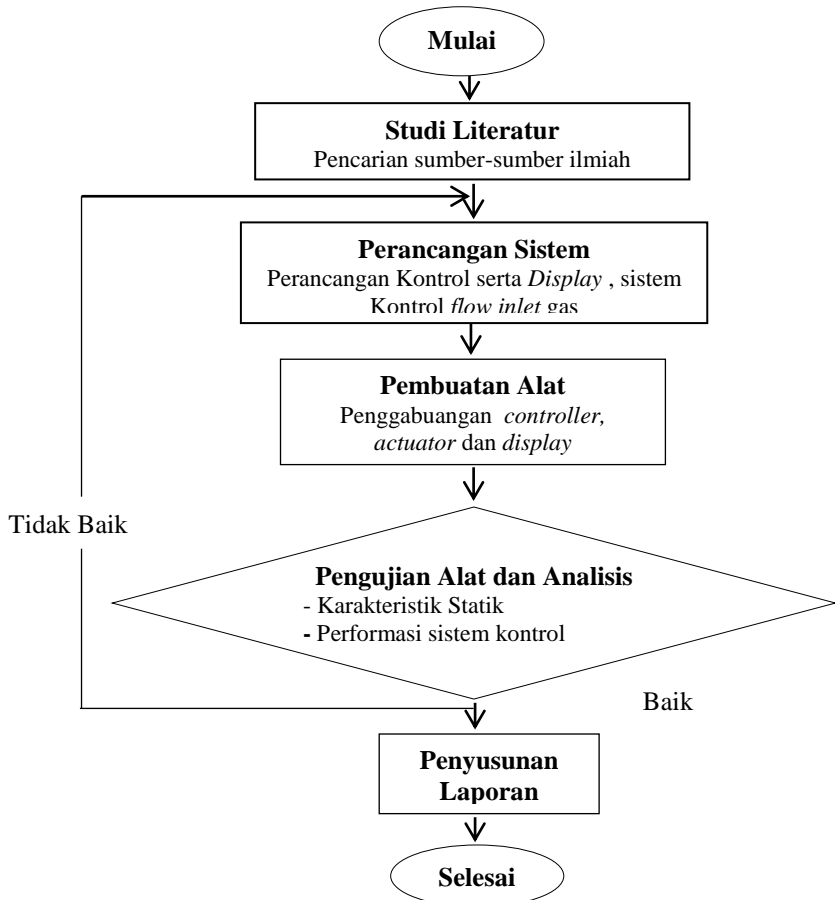
Gambar 2.10 Tabel T-student

BAB III

PERANCANGAN DAN PEMBUATAN ALAT

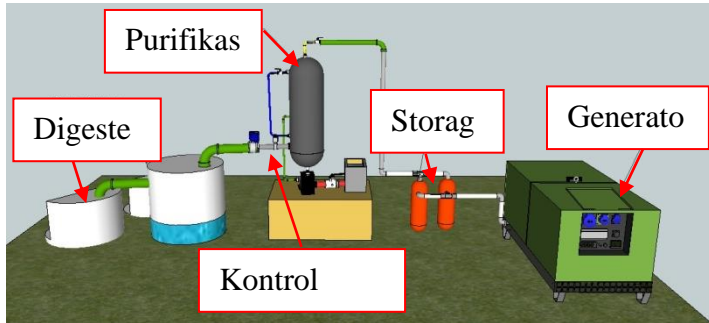
3.1. Diagram Alir (*Flowchart*)

Langkah-langkah perancangan alat ini digambarkan dalam *flowchart* penelitian yang dapat dilihat pada gambar berikut ini :



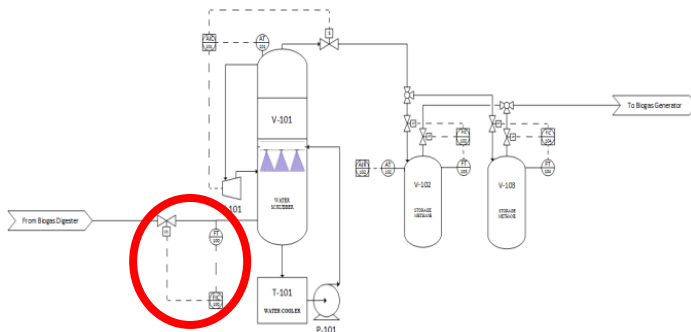
Gambar 3.1 *Flowchart* Tugas Akhir

Desain Plat purifikasi biogas dibuat sedemikian rupa dimana perancangan control flow rate inlet pada kolom purifikasi biogas yang dibuat. Berikut merupakan desain plant purifikasi biogas.



Gambar 3.2 Desain *Plant* Purifikasi Biogas

Dari gambar diatas maka dapat dibuat sebuah *piping and instrumentation diagram* pada plant purifikasi biogas seperti pada gambar dibawah ini.



Gambar 3.3 P&id Purifikasi Biogas

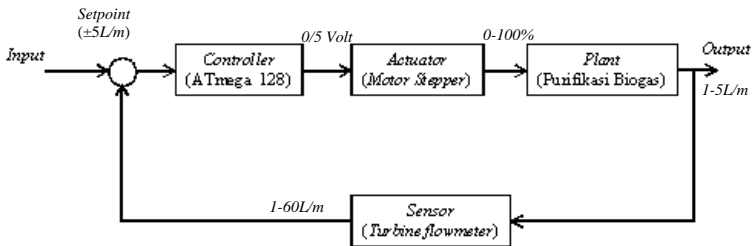
3.2. Studi Literatur

Mempelajari literatur dari penelitian – penelitian yang pernah dilakukan sebelumnya mengenai Pengendalian Flow Inlet Gas pada Alat Purifikasi Biogas. Dari hasil studi literatur yang telah dilakukan, didapatkan bahwa pengendalian flow inlet

mempengaruhi hasil pemurnian biogas dengan menggunakan *setpoint* laju Aliran sebesar $\pm 7,41 \text{ Nm}^3/\text{h}$ akan tetapi laju tersebut tergantung dari hasil produksi biogas sehingga *setpoint* yang sesuai pada Tugas Akhir ini sebesar $\pm 5 \text{ L/m}$ yang dapat dideteksi menggunakan sensor *turbine flowmeter* ukuran 1 inch dengan penggerak menggunakan *motor stepper*.

3.3. Perancangan Sistem

Pada tahap ini di lakukan perancangan sistem pengendalian laju aliran pada flow inlet purifikasi biogas. Perancangan sistem akan menjelaskan tentang perancangan-perancangan elektrik seperti rangkaian pengkondisian sinyal dan perancangan mekanik alat. Berikut ini adalah diagram blok dari Pengendalian Flow Inlet Gas pada Alat Purifikasi Biogas dengan *Water Scrubber System*.



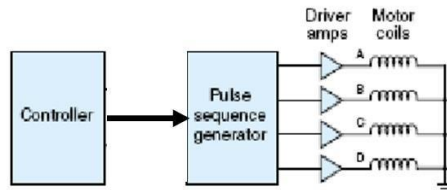
Gambar 3.4 Diagram Blok Pengendalian Flow Inlet

Diagram blok tersebut menjelaskan mengenai alur pengendalian yang dilakukan oleh sensor. Mulai dari input, apabila sensor mendeteksi terjadinya error ketika laju aliran gas tidak sesuai dengan set point, sinyal akan diproses oleh controller yang dapat menggerakkan actuator. Actuator ini lah yang akan mengatur laju aliran. Aktuator akan meneruskan gas yang sudah memenuhi setpoint menuju tabung purifikasi.

- Perancangan Motor Stepper

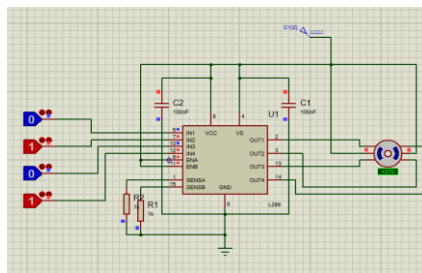
Perancangan motor stepper dibutuhkan rangkaian driver motor stepper untuk memberikan energi (energizing) pada pasangan-pasangan kumparan lebih

besar dari kemampuan I/O port, terdapat dimana rangkaian ini berisi 4 rangkaian yang diparalel. Setiap 1 rangkaian dihubungkan ke masing – masing kutub kumparan pada motor stepper. Motor stepper pada plant ini dipasang pada gate valve yang berguna sebagai aktuatur pengendalian flow.



Gambar 3.5 Diagram blok rangkaian driver motor stepper

Rangkaian pada Gambar 3.9 ini pada dasarnya hanya merupakan rangkaian switching arus yang mengalir lilitan pada motor stepper. Urutan pemberian data pada motor stepper ini dapat mengontrol arah putaran dari motor stepper ini. Penambahan kecepatan pada motor stepper dapat dilakukan dengan cara meningkatkan frekuensi pemberian data pada rangkaian switching arus. Rangkaian driver motor stepper ini nantinya terhubung langsung dengan lilitan pada motor, rangkaian power suplai, dan rangkaian mikrokontroler yang pada akhirnya menentukan kapan lilitan yang diinginkan dalam kondisi off atau on.

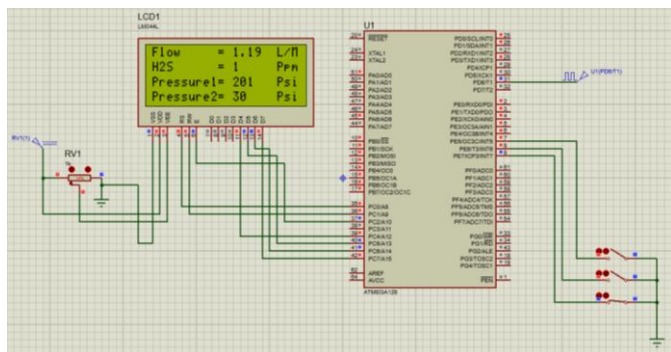


Gambar 3.6 Simulasi Motor Stepper

Pada mode operasi full step, kedua kumparan pada motor stepper dibangkitkan dengan 2 sinyal seperti pada Gambar 3.9 Apabila sinyal ini dibagi untuk setiap pulsa clock maka terdapat 4 fase yang setiap fasenya akan menghasilkan rotasi motor sebesar $1,8^\circ$. Mode operasi full step merupakan mode operasi yang menggerakkan rotor pada nilai sudut tertentu dalam satu buah sinyal clock yang diinputkan. Ketika dilihat dari segi respon, full step memiliki respon yang cepat. Hal ini dikarenakan nilai pwm yang digunakan yaitu hanya 0 dan 255 sehingga untuk melakukan rotasi sebesar 360° membutuhkan 200 step ($360^\circ / 1,8^\circ$ (spesifikasi pada motor stepper)). Pada metode full step ini untuk menempuh sudut $1,8^\circ$ hanya melakukan 1 kali step.

- Perancangan Tampilan LCD 20x4

Modul LCD (*Liquid Crystal Display*) yang digunakan dalam tugas akhir ini adalah modul LCD yang dapat menampilkan 20x4 karakter, yaitu 20 karakter untuk kolom dan 4 karakter untuk baris.



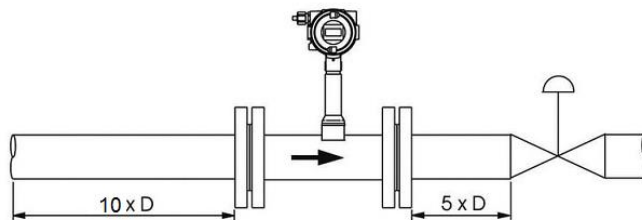
Gambar 3.5 Rangkaian LCD 20x4

Pada gambar rangkaian 3.15 tersebut, kaki-kaki dari rangkaian LCD dihubungkan ke port C mikrokontroler. Pin 1 dihubungkan ke Vcc (5V), pin 2 dan 16

dihubungkan ke Gnd (*Ground*), pin 3 merupakan pengaturan tegangan *Contrast* dari LCD, pin 4 merupakan *Register Select* (RS), pin 5 merupakan R/W (*Read/Write*), pin 6 merupakan *Enable*, pin 11-14 merupakan data. *Reset*, *enable*, R/W dan data dihubungkan ke mikrokontroler ATmega128. Fungsi dari potensiometer dalam rangkaian 3.14 tersebut adalah untuk mengatur gelap/terangnya (*brighthness*) karakter yang ditampilkan pada LCD. Rangkaian LCD ini digunakan sebagai penampil data *flow-inlet* pada purifikasi biogas.

- Perancangan Peletakan Mekanik Alat

Dalam pemasangan *flowmeter* dan MOV ini tidak boleh dilakukan secara sembarangan. Terdapat standar yang harus dilakukan dalam pemasangannya yaitu jarak pemasangan antar *instrument* ini. Hal ini sangat penting untuk dilakukan agar sistem pengendalian *flow* dapat berjalan dengan baik dan terhindar dari turbulensi. Adapun standar jarak pemasangan *flowmeter* dan MOV sesuai dengan gambar 3.4.



Gambar 3.8 Standar installasi pemasangan *flowmeter*

Dari gambar 3.4, diketahui standar pemasangan *flow meter* dari *control valve* yaitu $5 \times$ diameter pipa.

$$X = 5 \times \text{Diameter Pipa}$$

Dimana plant ini menggunakan pipa berukuran 1 inch (2,54 cm). Maka panjang X pada plant yaitu :

Panjang X (jarak antara *water flowmeter* dan *valve*)

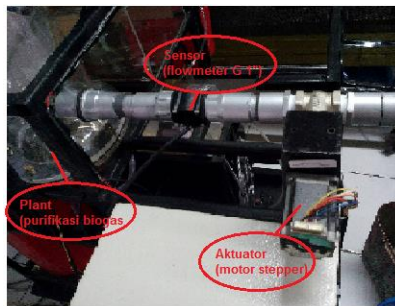
$$X = 5 \times 2,54 \text{ cm}$$

$$X = 12,7 \text{ cm}$$

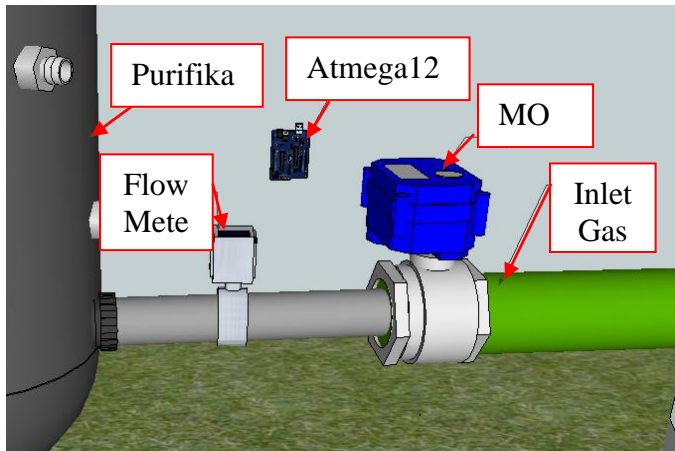
3.4. Pembuatan Alat

Pada tahap pembuatan alat ini akan dibagi menjadi 2 bagian, yaitu pembuatan hardware alat dan pembuatan software alat. Untuk pembuatan hardware terdiri dari pembuatan elektik dan mekanik pada alat, sedangkan pembuatan software berisi langkah-langkah cara mengkodng mikrokontroler agar sistem control dapat berjalan sesuai dengan yang diinginkan.

- Pembuatan Hardware Alat



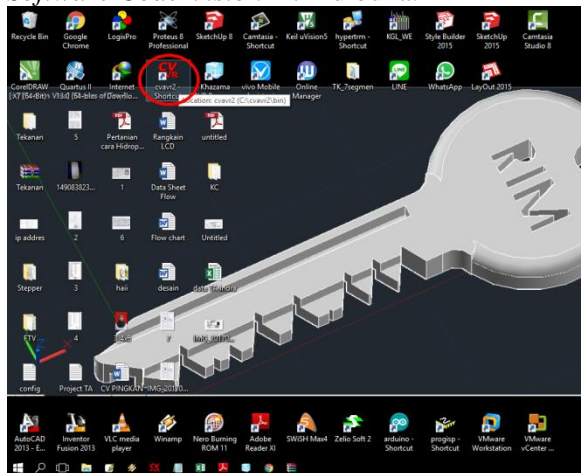
Gambar 3.9 Realisasi Pemasangan Sistem Kontrol *Flow*



Gambar 3.10 Desain Sistem Kontrol *Flow*

- Pembuatan Software Alat
Kemudian untuk pemrograman menggunakan *software Code Vision AVR* dilakukan dengan langkah-langkah sebagai berikut:

a. *Software Code Vision AVR* dibuka.



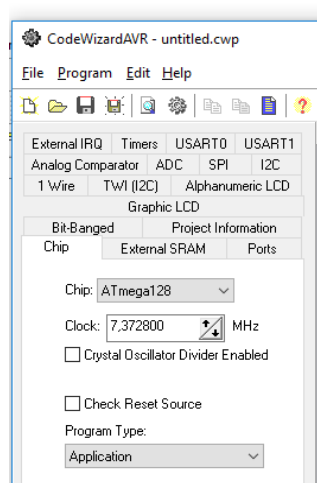
Gambar 3.11 Buka *Software*

- b. *File* dipilih, kemudian *New* dipilih. Kemudian akan muncul gambar seperti dibawah ini.



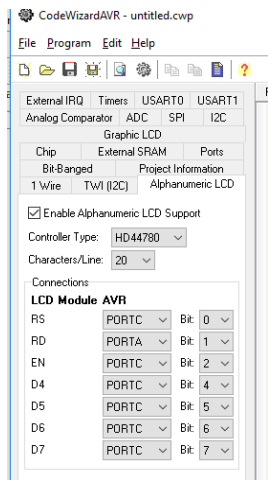
Gambar 3.12 *Create New File*

- c. Pada *File Type*, *Project* dipilih, kemudian OK dipilih.
- d. Pada *Chip*, ATmega128 dipilih dan pada frekuensi *clock* diisi seperti gambar 3.13.



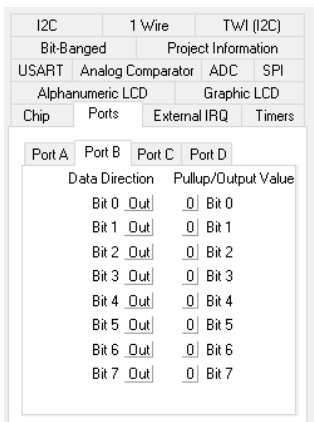
Gambar 3.13 *Chip pada Code Vision AVR*

- e. Pada *Alphanumeric LCD*, *Enable Alphanumeric LCD Support* dipilih, kemudian pada *Characters/Line* dipilih 20, dan pada *LCD Module AVR* dipilih *PORTC*.



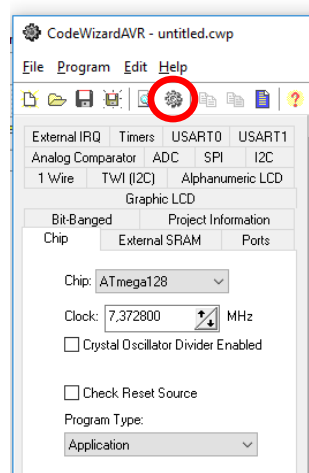
Gambar 3.14 *Alphanumeric LCD* pada *Code Vision AVR*

- f. Pada *Ports*, *PORTB* dipilih dan semua *pin* dijadikan *output*.



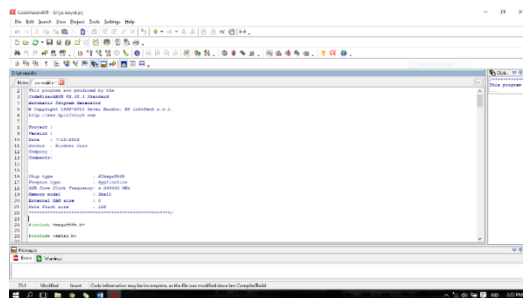
Gambar 3.15 konfigurasi pada *PORTB*

- g. Kemudian tombol seperti dibawah ini dipilih.



Gambar 3.16 Tombol *Generate*

- h. Setelah itu akan muncul program yang telah diatur dan pemrograman bisa dilakukan.

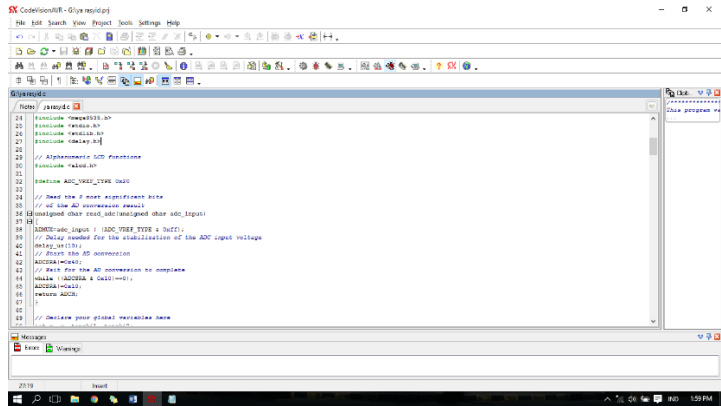


Gambar 3.17 Tampilan Awal Program

- i. Setelah selesai menyusun program, program dapat di-*compile*.

Pembacaan nilai *flowrate* akan ditampilkan pada LCD 4x20 sebagai *display*. Pada *plant purifikasi biogas* ini

digunakan *set point* 5L/m, dengan *motor stepper* sebagai aktuator.



```

// Nama : asyudic
24 #include <math.h>
25 #include <stdio.h>
26 #include <stdlib.h>
27 #include <string.h>
28 // Alphanumeric I/O functions
29 #include <ctype.h>
30 #include <conio.h>
31 #include <iomanip>
32 #include <iostream>
33 // Read the 8 most significant bits
34 // of the AD conversion result
35 // Assigned char read, also assigned char set, input
36 //
37 #define ADC_INPUT ( (ADC_TYPE & 0xFF) )
38 // Delay needed for the stabilization of the ADC input voltage
39 #define DELAY_100 (100)
40 // Stop the AD conversion
41 #define STOP_ADC (0)
42 #define START_ADC (1)
43 // While for the AD conversion no complete
44 #define WHILE_ADC ( (ADC_RESULT & 0xFF) == 0 )
45 #define END_ADC (1)
46 #define END_ADC (1)
47 // Define your global variables here
48
// Notes
//
//
//

```

Gambar 3.18 Tampilan *Listing Program Sistem*

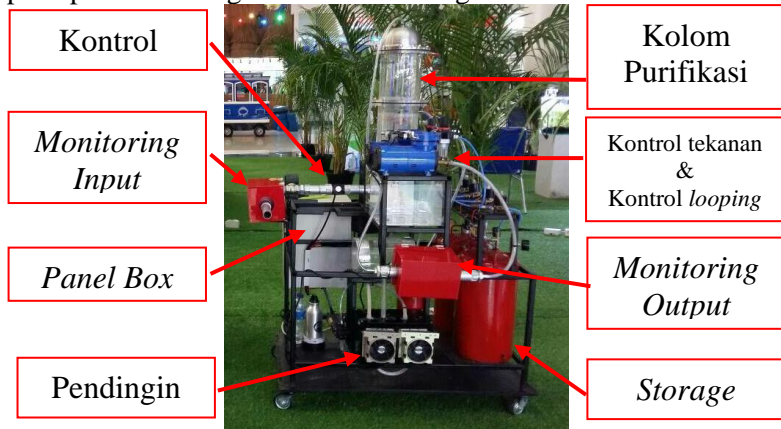
3.5. Pengujian Alat dan Analisis

Pada tahap pengujian alat ini dilakukan percobaan menggunakan kompresor sebagai laju aliran input dan dibandingkan dengan rotameter sebagai alat yang sudah standart untuk mengetahui nilai range, span, histerisis, akurasi serta kesalahan yang ada pada alat tersebut. Analisa Data dilakukan dengan cara melakukan pengambilan data dan data tersebut diolah agar didapatkan nilai sesuai dengan karakteristik maupun rumusan yang telah ditentukan.

BAB IV HASIL DATA DAN PEMBAHASAN

4.1 Rancang Bangun Alat

Berikut ini adalah perancangan sistem kontrol *flow inlet* pada purifikasi biogas berbasis ATmega128.



Gambar 4.1 *Purification System Plant*

Dari hasil biodigester biogas, maka gas akan diteruskan pada purifikasi untuk penghilangan gas pengotor lainnya yang dapat menurunkan kualitas dari biogas tersebut. Agar didapatkan tingkat efektifitas dan efisiensi yang lebih tinggi maka sebelum memasuki kolom purifikasi terlebih dahulu dilakukan sistem *control flowrate* pada aliran biogas. Proses *control flowrate* ini menggunakan sensor flowmeter dengan actuator berupa motor stepper yang dapat mengendalikan aliran pada inlet purifikasi sebelum memasuki kolom purifikasi dengan setpoint laju aliran sebesar 5 L/min.

4.2 Pengujian Alat

Sensor tekanan *flowmeter* pengujian sensor dilakukan dengan memberikan flowrate menggunakan kompresor. Hasil pembacaan sensor dapat dilihat seperti pada tabel 4.1. pengambilan data dilakukan setiap 5 detik. Actuator motor

setteper dapat memberikan output mendekati setpoint sensor sebesar 5 L/min. Pengujian dilakukan untuk mengetahui grafik respon actuator dan pembacaan flowrate pada sensor berdasarkan waktu yang telah ditentukan.

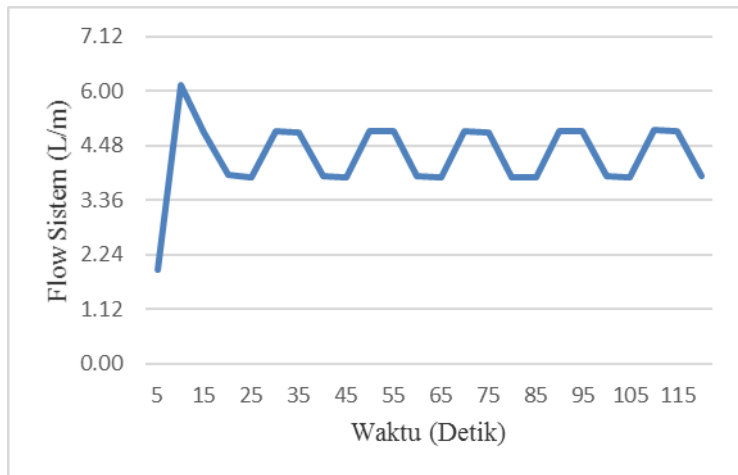
Tabel 4.1 Hasil Pengujian Sistem

No	Menit ke-	Detik Ke-	Flow Sistem
1	1	5	2.05
2		10	6.08
3		15	5.06
4		20	4.09
5		25	4.06
6		30	5.07
7		35	5.06
8		40	4.08
9		45	4.06
10	1	50	5.07
11		55	5.07
12		60	4.08
13	2	65	4.07
14		70	5.08
15		75	5.06
16		80	4.07
17		85	4.06
18		90	5.08
19		95	5.08
20		100	4.08
21		105	4.07

Tabel 4.1 Hasil Pengujian Sistem (Lanjutan)

No	Menit ke-	Detik Ke-	Flow Sistem
22		110	5.09
23		115	5.07
24		120	4.08

Dari data tersebut maka diperoleh sebuah grafik pengujian sensor sebagai berikut.

**Gambar 4.2** Grafik Respon sistem

Dari grafik diatas adalah data flowrate yang di deteksi oleh sensor dengan setpoint 5 L/min. aliran biogas yang di kendalikan oleh motor stepper dapat bergerak dengan menyesuaikan setpoint yang telah ditentukan.

4.3 Pengukuran Kalibrasi Sensor

Kalibrasi sensor ini sangat diperlukan agar keluaran sensor dapat sesuai dengan yang sudah ditetapkan atau standart. Berikut merupakan data yang didapatkan dari hasil kalibrasi.

Tabel 4.2 Hasil Pengambilan Data Naik

No	Pembacaan Standart	Data ke					Rata-Rata
		1	2	3	4	5	
1	0	0	0	0	0	0	0
2	1	1	1	1.2	1	1.1	1.06
3	3	3.2	3	3.3	3	3	3.1
4	4	4.4	4.2	4.4	4	4	4.2
5	5	5.3	5.5	5	5	5.1	5.18
Jumlah							13.54
Rata-rata							2.708

Tabel 4.3 Hasil Pengambilan Data Turun

No	Tekanan Standart (L/m)	Data ke					Rata-Rata
		1	2	3	4	5	
1	0	0	0	0	0	0	0
2	1	1.4	1	1.2	1	1	1.12
3	3	3.5	3.1	3	3	3	3.12
4	4	4.7	4.2	4.4	4	4.6	4.38
5	5	5.3	5.7	5	5	5.1	5.22
Jumlah							13.84
Rata-rata							2.768

Tabel 4.4 Data Kalibrasi Sensor a

No	Pembacaan Std (X)	Pembacaan Alat		Rata- rata	Koreksi (Y)
		Rata- rata Naik	Rata- rata turun		
1	0	0	0	0	0
2	1	1.06	1.12	1.09	-0.09
3	3	3.1	3.12	3.11	-0.11
4	4	4.2	4.38	4.29	-0.29
5	5	5.18	5.22	5.2	-0.2
Jumlah					-0.138
Rata- rata					-0.69

Tabel 4.5 Data Kalibrasi Sensor b

No	X	X ²	Y-Y'	X*Y	(Y-Y') ²
1	0	0	0.138	0	0.019044
2	1	1	0.048	-0.09	0.002304
3	3	9	0.028	-0.33	0.000784
4	4	16	-0.152	-1.16	0.023104
5	5	25	-0.062	-1	0.003844
Jumlah	13	51	-2.58	0.04908	-8.31163
Rata- rata	2.6	10.2	-0.516	0.009816	-1.66233

Berikut ini merupakan cara untuk mencari nilai ketidakpastian sensor MPX5500DP 1.

a. Nilai Ketidakpastian Tipe A

$$\sigma = \frac{\sqrt{\sum(y-\bar{y})^2}}{n-1} \dots\dots\dots (4.1)$$

$$\sigma = 0,11077$$

Sehingga didapatkan nilai U_{a1} sebagai berikut:

$$U_{a1} = \frac{\sigma}{\sqrt{n}} \dots \dots \dots (4.2)$$

$$U_{a1} = \frac{0,11077}{\sqrt{5}} = 0,0495$$

Sedangkan nilai ketidakpastian regresi $U_{a2} = \sqrt{\frac{SSR}{n-2}}$

Dimana:

SSR (*Sum Square Residual*) = $\sum SR$ (*Square Residual*)

$SR = R^2$ (*Residu*)

Y = Nilai koreksi

$$Y_{reg} = a + (b \cdot xi) \dots \dots \dots (4.3)$$

$$a = \bar{y}_i + (b \cdot \bar{x}) \dots \dots \dots (4.4)$$

$$b = \frac{n \sum xy - \sum x \sum y}{n \cdot \sum x^2 - (\sum x)^2} \dots \dots \dots (4.5)$$

$$SSR = \sum (y - Y_{reg})^2 \dots \dots \dots (4.6)$$

Dimana:

x = *Pemb. standar*,

y = *Nilai koreksi*, n = *Jumlah data*

$$b = \frac{(-12,9) - (-12,31)}{(255) - (169)} = -0,29314$$

Sehingga nilai:

$$a = (-0,138) + (-0,29314 \times 2,5)$$

$$a = -0,9$$

Jadi, persamaan regresi menjadi:

$$Y_{reg} = (-0,9) + (x(-0,29314))$$

Yang menghasilkan nilai **SSR = 12,68**

$$U_{a2} = \sqrt{\frac{SSR}{n-2}} = \sqrt{\frac{12,68}{5-2}} = 2,056$$

- b. Nilai Ketidakpastian *Type B* :

Pada tipe ini terdapat 2 parameter ketidakpastian, yaitu ketidakpastian Resolusi (U_{b1}) dan Ketidakpastian alat standar *pressure gauge* (U_{b2}). Dengan perhitungan sebagai berikut:

$$U_{b1} = \frac{\frac{1}{2} \times \text{Resolusi}}{\sqrt{3}} = \frac{\frac{1}{2} \times 1}{\sqrt{3}} = 0,2886751$$

$U_{b2} = \frac{a}{k}$, dikarenakan pada alat standar tidak ada sertifikat kalibrasinya maka nilai a (ketidakpastian sertifikat kalibrasi) dianggap mendekati 0, dan nilai faktor cakupan dianggap 2,0. Sehingga hasil : **$U_{b2} = 0$**

- c. Nilai ketidakpastian kombinasi U_c :

$$U_c = \sqrt{U_{a1}^2 + U_{a2}^2 + U_{b1}^2 + U_{b2}^2} \dots\dots\dots (4.7)$$

$$U_c = \sqrt{0,0495^2 + 12,056^2 + 0,2886^2 + 0^2}$$

$$U_c = 2,07696$$

4.4 Data Karakteristik Statik Sensor

Karakteristik statik merupakan sifat dari sebuah alat ukur yang tidak bergantung terhadap waktu. Berikut merupakan data perhitungan. Untuk mengetahui nilai untuk mengetahui karakteristik suatu alat.

Tabel 4.6 Hasil Data Sensor Naik dan Turun

No	FIn	Fout Naik	Fout Turun	H(I)	(Fin-Fout)/Fin
1	0	0	0	0	0
2	1	1.06	1.12	0.06	0.06
3	3	3.1	3.12	0.02	0.03
4	4	4.2	4.38	0.18	0.05
5	5	5.18	5.22	0.4	0.008
Jumlah					0.1513
Rata-rata					0.0302

Dari data yang terdapat dalam tabel maka dapat diketahui nilai karakteristik alat sebagai berikut:

- Range : 0 - 5 L/m
- Span : 5 L/m
- Histerisis : 7,66%
- Akurasi : 0,64%
- Error : 3,02%

Nilai tersebut dapat diketahui dengan menghitung menggunakan rumusan seperti berikut:

- Histerisis

$$\text{Histerisis } H(I) = O(I)_{\text{II}} - O(I)_{\text{IT}}, \hat{H} = H(I)_{\text{max}}$$

sehingga:

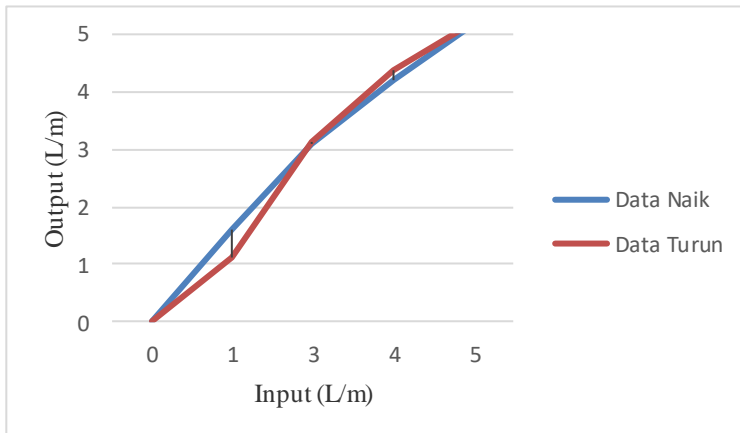
$$\% \text{ Maksimum histeresis} = \frac{\hat{H}}{O_{\text{max}} - O_{\text{min}}} \times 100\%$$

$$\% \text{ Maksimum histeresis} = \frac{0,4}{5,22 - 0} \times 100\%$$

$$\% \text{ Maksimum histeresis} = \frac{0,4}{5,22} \times 100\%$$

$$\% \text{ Maksimum histeresis} = 7,66 \%$$

Berikut gambar grafik histerisis dari data pengukuran naik dan turun tersebut:



Gambar 4.3 Grafik Histerisis Sensor

Pada gambar grafik diatas menunjukkan bahwa histerisis tertinggi terletak pada saat output 5 L/m.

b. Akurasi

$$A = 1 - \sum \left| \frac{Y_n - X_n}{Y_n} \right| \times 100\%$$

Dengan:

Y_n = Pembacaan standar (I) dan

X_n = Pembacaan alat (O)

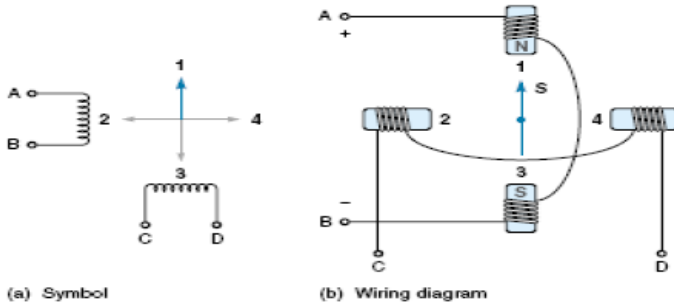
$$A = 20\% \times (0.032)$$

$$A = 0,64\%$$

4.5 Pengendalian Driver Motor Stepper pada MOV

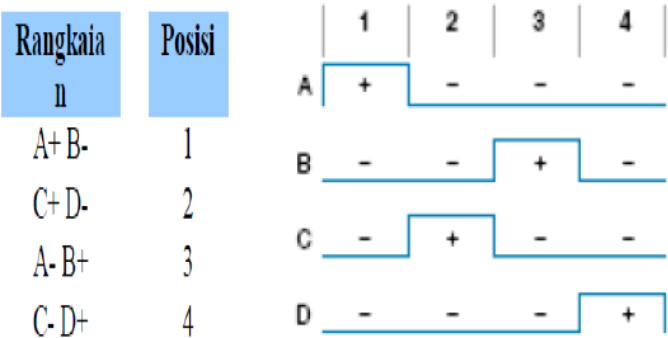
Penghubungkan motor stepper dengan piranti digital atau I/O port dibutuhkan rangkaian interface. Hal ini sangat penting karena jumlah arus yang diperlukan untuk memberikan energi (energizing) pada pasangan-pasangan kumparan lebih besar dari

kemampuan I/O port, sehingga dibutuhkan modul driver untuk dapat menggerakkan motor stepper. Berikut ini adalah ilustrasi struktur motor stepper sederhana dan pulsa yang dibutuhkan untuk menggerakkannya



Gambar 4.4 Motor stepper dua-phase (bipolar)

Penggunaan motor stepper jenis bipolar memerlukan rangkaian yang agak lebih rumit untuk mengatur agar motor ini dapat berputar dalam dua arah. Motor stepper dua-phase (bipolar) hanya mempunyai dua rangkaian tetapi sebenarnya terdiri dari empat kutub medan. Cara yang paling sederhana dalam memberikan step pada motor ini adalah dengan memberikan energi secara bergantian pada AB atau CD untuk menarik rotor dari kutub ke kutub. Jika rotor bergerak CCW (counterclockwise, berlawanan arah jarum jam) dari posisi 1, maka rangkaian CD harus diberi energi dengan polaritas C+D-. Hal ini akan menarik rotor ke posisi 2. Selanjutnya, rangkaian AB diberi energi lagi, tetapi kali ini polaritasnya terbalik (-A+B), yang menyebabkan kutub bawah memberikan ujung utara pada rotor, dengan demikian tertarik ke posisi 3. Istilah bipolar digunakan pada motor ini karena arus kadang-kadang terbalik. Urutan tegangan diperlukan untuk memutar motor satu putaran penuh dan ditunjukkan di bawah ini. Pembacaan dari atas ke bawah memberikan urutan untuk peralihan/perputaran CCW, pembacaan dari bawah ke atas adalah urutan CW (clockwise)



Gambar 4.5 Logika pengendalian motor stepper

Tabel kebenaran pengendalian MOV untuk membuka atau menutup valve sebagai berikut :

Tabel 4.7 Tabel kebenaran pengendalian motor stepper

Open Valve				Close Valve			
Orange	biru	kuning	merah	orange	biru	kuning	merah
0	0	0	1	1	0	0	0
0	1	0	0	0	0	1	0
0	0	1	0	0	1	0	0
1	0	0	0	0	0	0	1
0	0	0	1	1	0	0	0
0	1	0	0	0	0	1	0
0	0	1	0	0	1	0	0
1	0	0	0	0	0	0	1
dst.				dst.			

4.6 Pembahasan

Dilakukan pengujian pengendalian flow dengan menggunakan aktuator MOV. MOV diletakkan pada input sensor flow dan kolom purifikasi, sedangkan flowmeter dipasang pada output MOV dan input kolom purifikasi. Flowmeter yang digunakan yaitu flowmeter ukuran 1" dengan range kerja antara 1 liter/menit hingga 60 liter/menit. Rangkaian driver MOV menggunakan rangkaian modul driver L298 yang mampu bekerja hingga arus 4A, sedangkan beban motor yang digunakan yaitu 3A. Cara pengendalian MOV yaitu sesuai dengan Tabel 4.8 Dari kalibrasi yang dilakukan, diketahui error flowmeter sebesar 3,027% dengan maximum overshoot yaitu 6 liter/menit. Pengendalian aliran yang dilakukan yaitu mengatur input kolom purifikasi agar output sensor sesuai dengan setpoint yang ditentukan. Setpoint yang diinginkan yaitu 5 liter / menit. Metode pengendalian yang dilakukan yaitu saat pembacaan sensor flow kurang dari 5 Liter/menit, maka MOV akan bergerak membuka dan ketika pembacaan flow lebih dari 5 maka MOV akan bergerak menutup hingga flow mencapai setpoint yang ditentukan yaitu 5liter/ menit. waktu yang diperlukan MOV bergerak hingga setpoint flow tercapai yaitu kurang lebih 25 detik.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan rancang bangun dan analisa data yang telah dilakukan maka dapat disimpulkan bahwa :

- Terjadinya *osilasi* pada pembacaan flow pada pruiifikasi biogas dengan *setpoint* yang ditentukan yaitu 5 liter/menit menggunakan sensor *flowmeter* dan aktuator berupa *motor stepper*.
- Aktuator berupa motor stepper berjalan sesuai *setpoint* yang dinginkan yaitu saat pembacaan flow mendekati 5 liter/menit maka *motor stepper* akan berhenti. Data pengujian alat pada sistem pengendalian *flow*, didapatkan *respon time* yang cukup baik untuk mencapai *steady state* dengan maksimum *overshoot* 6 liter/menit dan waktu yang dibutuhkan untuk mencapai *set point* (*settling time*) dari awal pengambilan data adalah ± 25 detik

5.2 Saran

Adapun saran untuk penelitian selanjutnya agar hasil yang dicapai dapat memenuhi harapan, antara lain:

- Untuk menghindari *losse* saat pengendalian, dipastikan bahwa plant sudah tidak ada yang bocor karena akan mempengaruhi pengukuran *flow*
- Untuk menghindari *disturben* perhatikan jarak pemasangan *flowmeter* dari sumber turbulensi agar didapatkan pengukuran *flow* yang stabil.

(Halaman ini sengaja dikosongkan)

DAFTAR PUSTAKA

- [1] Kementerian ESDM, (2013), Supply Demand Energi. Pusat Data dan Teknologi Informasi Energi dan Sumber Daya Mineral Kementerian ESDM. Jakarta.
- [2] Osorio, F. & Torres, J.C., (2009), *Biogas Purification From Anaerobic Digestion In A Waste Water Treatment Plant for Biofuel Production*. Renewable Energy, 34, 2164-2171
- [3] Abdurrakhman, Arief, dkk. (2013), *Rancang Bangun Alat Purifikasi Biogas dengan Menggunakan CaO dan Water Scrubber*. Seminar Nasional Pascasarjana XIII – ITS. Surabaya.
- [4] Läntelä , J., Rasi, S., Lehtinen J., Rintala J., (2011), Landfill gas upgrading with pilot-scale water scrubber: Performance assessment with absorption water recycling. Published by Elsevier Ltd.
- [5] Curtis D. Johnson (1997), “Process Kontrol Instrumentation Technology” Fifth Edition, Prentice Hall International Inc., New Jersey.
- [6] Syahrul. 2016. Laporan Akhir. Jurusan Teknik Komputer Universitas Komputer Indonesia
- [7] Anonim. 2016. Brushless DC motor control using PLC School of Engineering, University of Western Sydney
- [8] Abdurrakhman, A., Soehartanto, T., (2014), Analisis Karakterisasi Water Scrubber pada Alat Purifikasi Biogas Tipe Kombinasi Spray Tower dan Tray Tower. Seminar Nasional Teknologi Terapan, Universitas Gadjah Mada, Yogyakarta.
- [9] Paguyuban KUD Jawa Timur, (2011), Laporan Biogas untuk Rumah Tangga (BIRU), Dinas Pertanian Provinsi Jawa Timur.
- [10] Hermanto., Arba Susanty., (2016), Pengaruh konsentrasi NaOH dan laju alir gas pada proses pemurnian biogas. Jurnal riset teknologi industry, Samarinda

LAMPIRAN A

(LISTING PROGRAM PADA CODE VISION AVR)

/******

This program was produced by the
CodeWizardAVR V2.05.3 Standard
Automatic Program Generator
© Copyright 1998-2011 Pavel Haiduc, HP InfoTech s.r.l.
<http://www.hpinfotech.com>

Project : TA_TEAM BIOGAS

Version :

Date : 08/06/2017

Author : RADIAN INDRA M

Company : ITS

Comments:

Chip type : ATmega128

Program type : Application

AVR Core Clock frequency: 7,372800 MHz

Memory model : Small

External RAM size : 0

Data Stack size : 1024

*** /

#include <mega128.h>

#include <delay.h>

#include <ds1307_twi.h>

// Standard Input/Output functions

#include <stdio.h>

#include <stdlib.h>

```
#define xmin 22
#define ymin 23
#define a 35      //pembacaan pressure sensor MPX5500DP
                  (max)
#define b 255    //8 bits (max)
#define e 8925   //a*b

// I2C Bus functions
#include <i2c.h>

#asm
    .equ __lcd_port=0x15 ;PORTC
#endasm
#include <lcd.h>

#ifndef RXB8
#define RXB8 1
#endif

#ifndef TXB8
#define TXB8 0
#endif

#ifndef UPE
#define UPE 2
#endif

#ifndef DOR
#define DOR 3
#endif

#ifndef FE
#define FE 4
#endif
```

```
#ifndef UDRE
#define UDRE 5
#endif
```

```
#ifndef RXC
#define RXC 7
#endif
```

```
#define FRAMING_ERROR (1<<FE)
#define PARITY_ERROR (1<<UPE)
#define DATA_OVERRUN (1<<DOR)
#define DATA_REGISTER_EMPTY (1<<UDRE)
#define RX_COMPLETE (1<<RXC)
```

```
// Write a character to the USART1 Transmitter
#pragma used+
void putchar1(char c)
{
    while ((UCSR1A & DATA_REGISTER_EMPTY)==0);
    UDR1=c;
}
#pragma used-
```

```
#define ADC_VREF_TYPE 0x20
```

```
// Read the 8 most significant bits
// of the AD conversion result
unsigned char read_adc(unsigned char adc_input)
{
    ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
    // Delay needed for the stabilization of the ADC input voltage
    delay_us(10);
    // Start the AD conversion
    ADCSRA|=0x40;
    // Wait for the AD conversion to complete
```

```

while ((ADCSRA & 0x10)==0);
ADCSRA|=0x10;
return ADCH;
}

```

```

// TWI functions

```

```

#include <twi.h>

```

```

// Declare your global variables here

```

```

//----- Gas Metana Input -----
=====ENDLY DEVIRA Y -2414031030-=====
unsigned long int temp1a,
temp1b,sensor1,vinMetan1,RL1,Rasio1,RS1,RO1,ppm1,
og1,ppm1,pangkat,e1;
unsigned char tampil1 [5] ;
//----- Gas Metana Output -----
unsigned long int temp2a, temp2b, sensor2;
unsigned char tampil2 [5] ;
//----- Gas CO2 Input -----
unsigned long int temp3a, temp3b, sensor3, ADC, range,
linierisasi, ppmCO2, vinco2;
unsigned char tampil3 [5] ;
//----- Gas CO2 Output -----
unsigned long int temp4a, temp4b, sensor4, ADCa, rangea,
linierisasia, ppmCO2a, vinco2a;
unsigned char tampil4 [5] ;
//----- Gas H2S Input -----
unsigned long int sensor5;
float VinH2S2, RS2, RO2,hsl_pangkat2,R2,pangkat2,ppm2;
char buf6[33];
//----- Gas H2S Output -----
unsigned long int sensor6;
float VinH2S3, RS3, RO3,hsl_pangkat3,R3,pangkat3,ppm3;
char buf7[33];
//----- Real Time Clock -----

```



```

unsigned char buf0 [17];
unsigned char buf1 [17];
unsigned char s,m,h;
unsigned char dd,mm,yy;

```

```

//-----Sensor    FLow-----
=====RADIAN  INDRA  MUKROMIN  -2414031034-
=====

```

```

float freq;          // to store value of frequency value
unsigned int i=0,countx;
//i=number of overflows in one second
unsigned dur;        // dur to store the value of TCNT1 register
char buf2[8];
// the frequency value as a string to be displayed on lcd

```

```

//-----Sensor    Tekanan-----
=====LAYLY DIAN E -2414031005-=====

```

```

unsigned long int sensor1, sensor2, tekanan1, tekanan2;
float x,y;
float c1,c2,d1,d2,f1,f2,g1,g2,h1,h2;
float l=0;
float j=0;
float k=0;
float tangki1, tangki2;
char buf3[33];
char buf4[33];

```

```

//-----Sensor gas H2s Kontrol-----
-----=====EKA WAHYU P -24140310XX-=====

```

```

unsigned long int sensor3;
float VinH2S1, RS0, RO0,hs_pangkat,R0,pangkat0,ppm0;
char buf5[33];

```

```

void tampil_LCD(void);
void devi (void);
void indra (void);

```

```

void layli (void);
void wahyu (void);
void date(void);
void berotasi(unsigned char b3, unsigned char b2,unsigned
               char b1, unsigned char b0);
void startup(void);
void cond_1(void);
void cond_2(void);
void cond_3(void);
void cond_4(void);
void cond_5(void);
void cond_6(void);
void cond_7(void);

```

```

/*****

```

```

Syntak_1(PEMBACAAN FLOW)

```

```

sistem Pengendalian flow inlet pada kolom purifikasi

```

```

====RADIAN INDRA MUKROMIN====

```

```

-2414031034-

```

```

*****/

```

```

// Timer1 overflow interrupt service routine

```

```

interrupt [TIM1_OVF] void timer1_ovf_isr(void)

```

```

{

```

```

// Place your code here

```

```

i++; // count the number of overflows in one second

```

```

}

```

```

void main(void)

```

```

{

```

```

// Declare your local variables here

```

```

// Input/Output Ports initialization

```

```

// Port A initialization

```

```

// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out

```

```

    Func2=Out Func1=Out Func0=Out

```

```

// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0
    State1=0 State0=0
PORTA=0x00;
DDRA=0xFF;

// Port B initialization
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=In
    Func2=Out Func1=In Func0=In
// State7=0 State6=0 State5=0 State4=0 State3=T State2=0
    State1=T State0=T
PORTB=0x00;
DDRB=0xFF;

// Port C initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In
    Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T
    State1=T State0=T
PORTC=0x00;
DDRC=0x00;

// Port D initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In
    Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T
    State1=T State0=T
PORTD=0x00;
DDRD=0x00;

// Port E initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In
    Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T
    State1=T State0=T
PORTE=0xFF;

```

```
DDRE=0x00;
```

```
// Port F initialization
```

```
// Func7=In Func6=In Func5=In Func4=In Func3=In  
    Func2=In Func1=In Func0=In
```

```
// State7=T State6=T State5=T State4=T State3=T State2=T  
    State1=T State0=T
```

```
PORTF=0x00;
```

```
DDRF=0x00;
```

```
// Port G initialization
```

```
// Func4=In Func3=In Func2=In Func1=In Func0=In
```

```
// State4=T State3=T State2=T State1=T State0=T
```

```
PORTG=0x00;
```

```
DDRG=0x00;
```

```
// Timer/Counter 0 initialization
```

```
// Clock source: System Clock
```

```
// Clock value: Timer 0 Stopped
```

```
// Mode: Normal top=0xFF
```

```
// OC0 output: Disconnected
```

```
ASSR=0x00;
```

```
TCCR0=0x00;
```

```
TCNT0=0x00;
```

```
OCR0=0x00;
```

```
// Timer/Counter 1 initialization
```

```
// Clock source: System Clock
```

```
// Clock value: Timer1 Stopped
```

```
// Mode: Normal top=0xFFFF
```

```
// OC1A output: Discon.
```

```
// OC1B output: Discon.
```

```
// OC1C output: Discon.
```

```
// Noise Canceler: Off
```

```
// Input Capture on Falling Edge
```

```
// Timer1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
// Compare C Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x07; //RADIAN INDRA-untuk mengatur timer
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;
OCR1CH=0x00;
OCR1CL=0x00;
```

```
// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer2 Stopped
// Mode: Normal top=0xFF
// OC2 output: Disconnected
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;
```

```
// Timer/Counter 3 initialization
// Clock source: System Clock
// Clock value: Timer3 Stopped
// Mode: Normal top=0xFFFF
// OC3A output: Discon.
// OC3B output: Discon.
// OC3C output: Discon.
// Noise Canceler: Off
```

```
// Input Capture on Falling Edge
// Timer3 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
// Compare C Match Interrupt: Off
TCCR3A=0x00;
TCCR3B=0x00;
TCNT3H=0x00;
TCNT3L=0x00;
ICR3H=0x00;
ICR3L=0x00;
OCR3AH=0x00;
OCR3AL=0x00;
OCR3BH=0x00;
OCR3BL=0x00;
OCR3CH=0x00;
OCR3CL=0x00;
```

```
// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
// INT3: Off
// INT4: Off
// INT5: Off
// INT6: Off
// INT7: Off
EICRA=0x00;
EICRB=0x00;
EIMSK=0x00;
```

```
// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x04; //RADIAN INDRA-untuk mengatur timer1
```

ETIMSK=0x04; //RADIAN INDRA-untuk mengatur timer

// USART0 initialization

// USART0 disabled

UCSR0B=0x00;

// USART0 initialization

// Communication Parameters: 8 Data, 1 Stop, No Parity

// USART0 Receiver: On

// USART0 Transmitter: On

// USART0 Mode: Asynchronous

// USART0 Baud Rate: 9600

UCSR0A=0x00;

UCSR0B=0x18;

UCSR0C=0x06;

UBRR0H=0x00;

UBRR0L=0x2F;

// Analog Comparator initialization

// Analog Comparator: Off

// Analog Comparator Input Capture by Timer/Counter 1: Off

ACSR=0x80;

SFIOR=0x00;

// ADC initialization

// ADC Clock frequency: 921,600 kHz

// ADC Voltage Reference: AREF pin

// Only the 8 most significant bits of

// the AD conversion result are used

ADMUX=ADC_VREF_TYPE & 0xff;

ADCSRA=0x83;

// SPI initialization

// SPI disabled

```

SPCR=0x00;

// TWI initialization
// Mode: TWI Master
// Bit Rate: 100 kHz
twi_master_init(100);

// I2C Bus initialization
// I2C Port: PORTE
// I2C SDA bit: 2
// I2C SCL bit: 3
// Bit Rate: 100 kHz
// Note: I2C settings are specified in the
// Project|Configure|C Compiler|Libraries|I2C menu.
i2c_init();

// Alphanumeric LCD initialization
// Connections are specified in the
// Project|Configure|C Compiler|Libraries|Alphanumeric LCD
// menu:
// RS - PORTC Bit 0
// RD - PORTC Bit 1
// EN - PORTC Bit 2
// D4 - PORTC Bit 4
// D5 - PORTC Bit 5
// D6 - PORTC Bit 6
// D7 - PORTC Bit 7
// Characters/line: 20
lcd_init(20);
    //send Data ke Data Logger JUDUL-HEADER
    printf("TANGGAL,");
    delay_ms(10);
    printf("JAM,");
    delay_ms(10);
    printf("CH4/I,");

```



```

delay_ms(10);
printf("CH4/0,");
delay_ms(10);
printf("H2S/I,");
delay_ms(10);
printf("H2S/O,");
delay_ms(10);
printf("CO2/I,");
delay_ms(10);
printf("CO2/O,");
delay_ms(10);
printf("FLOW,");
delay_ms(10);
printf("H2S/C,");
delay_ms(10);
printf("TEKANAN 1,");
delay_ms(10);
printf("TEKANAN 2\n");
delay_ms(10);

//   rtc_set_time(10,00,00); //set time 10:00:00
//   rtc_set_date(0,11,06,17);

#asm("sei")

while (1)
{
    // Place your code here

    devi ();
    indra ();
    layli ();
    wahyu ();
    tampil_LCD();
    date()

```

```

//send Data ke Data Logger
printf("%02i:%02i:%02i",dd,mm,yy); //Tanggal
delay_ms(10);
printf(",");
printf("%02i:%02i:%02i",h,m,s); //jam
delay_ms(10);
printf(",");
printf("%i",sensor1); //CH4 Input
delay_ms(10);
printf(",");
printf("%i",sensor2); //CH4 Output
delay_ms(10);
printf(",");
printf("%.1f",hsl_pangkat2); //H2S Input
delay_ms(10);
printf(",");
printf("%.1f",hsl_pangkat3); //H2S Output
delay_ms(10);
printf(",");
printf("%i",sensor3); //CO2 Input
delay_ms(10);
printf(",");
printf("%i",sensor4); //CO2 Output
delay_ms(10);
printf(",");
printf("%.1f",freq); //Flow
delay_ms(10);
printf(",");
printf("%.1f",hs_pangkat); //H2S Control
delay_ms(10);
printf(",");
printf("%.1f",h1); //tekanan 1
delay_ms(10);
printf(",");
printf("%.1f",h2); //tekanan 2

```

```

    printf("\n");
}

}

void devi ()
{
    delay_ms(10);

//    //Metana input
    i2c_start(); // Start Condition
    i2c_write(0xE0); // Write to DT-SENSE module
    i2c_write(0x41); // "Read Sensor" Command
    i2c_stop(); // Stop Condition
    delay_us(10); // 10 us delay
    i2c_start(); // Start Condition
    i2c_write(0xE1); // Read from DT-SENSE module
    temp1a = i2c_read(1); // Data Sensor MSB
    temp1b = i2c_read(0); // Data Sensor LSB
    i2c_stop(); // Stop Condition
    sensor1 = (temp1a * 256) + temp1b ;
    itoa(sensor1,tampil1);
    delay_ms(10);

//    //Metana output
    i2c_start(); // Start Condition
    i2c_write(0xE2); // Write to DT-SENSE module
    i2c_write(0x41); // "Read Sensor" Command
    i2c_stop(); // Stop Condition
    delay_us(10); // 10 us delay
    i2c_start(); // Start Condition
    i2c_write(0xE3); // Read from DT-SENSE module
    temp2a = i2c_read(1); // Data Sensor MSB
    temp2b = i2c_read(0); // Data Sensor LSB
    i2c_stop(); // Stop Condition

```

```

    sensor2 = (temp2a * 256) + temp2b ;
    itoa(sensor2,tampil2);
    delay_ms(10);
//    sensor2=read_adc(6);
//    itoa(sensor2,tampil2);

//    //Co2 input
    i2c_start(); // Start Condition
    i2c_write(0xE4); // Write to DT-SENSE module
    i2c_write(0x41); // “Read Sensor” Command
    i2c_stop(); // Stop Condition
    delay_us(10); // 10 us delay
    i2c_start(); // Start Condition
    i2c_write(0xE5); // Read from DT-SENSE module
    temp3a = i2c_read(1); // Data Sensor MSB
    temp3b = i2c_read(0); // Data Sensor LSB
    i2c_stop(); // Stop Condition
    sensor3 = (temp3a * 256) + temp3b ;
//    vinco2 = (sensor3)*5/1203;
//    ADC= sensor3 - 1023;
//    range= 10000-350;
//    linierisasi= (1023-sensor3)/(10000-350);
//    ppmCO2= 350+linierisasi;
    itoa(sensor3,tampil3);
    delay_ms(10);

//
//    //Co2 output
    i2c_start(); // Start Condition
    i2c_write(0xE6); // Write to DT-SENSE module
    i2c_write(0x41); // “Read Sensor” Command
    i2c_stop(); // Stop Condition
    delay_us(10); // 10 us delay
    i2c_start(); // Start Condition
    i2c_write(0xE7); // Read from DT-SENSE module

```

```

temp4a = i2c_read(1); // Data Sensor MSB
temp4b = i2c_read(0); // Data Sensor LSB
i2c_stop(); // Stop Condition
sensor4 = (temp4a * 256) + temp4b ;
itoa(sensor4,tampil4);
delay_ms(10);

```

```

//sensor H2s Input
sensor5=read_adc(3);
VinH2S2= ((sensor5)*5)/255;
RO2 =4.38;
RS2 = ((5-VinH2S2)/VinH2S2)*10;
R2=RS2/RO2;
pangkat2= 1/(R2*R2*R2); //R0 di pangkat -3
ppm2 = 40*pangkat2;
hsl_pangkat2 = sensor5;

```

```

//sensor H2s Output
sensor6=read_adc(4);
VinH2S3= ((sensor3)*5)/255;
RO3 =4.38;
RS3 = ((5-VinH2S3)/VinH2S3)*10;
R3=RS3/RO3;
pangkat3= 1/(R3*R3*R3); //R0 di pangkat -3
ppm3 = 40*pangkat3;
hsl_pangkat3 = sensor6;
}

```

```

/*****

```

Syntax_2

sistem Pengendalian flow inlet pada kolom purifikasi

====RADIAN INDRA MUKROMIN -2414031034- =====

```

*****/

```

```

void indra ()
{
    //Pembacaan sensor Flow
    TIMSK=0x04;
    TCCR1B=0x07;
    delay_ms(1000);
    TCCR1B=0x00;
    TIMSK=0x00;
    dur=TCNT1;
    freq = (((dur + i*65536)*60)/4.8)*0.00226;
    TCNT1=0x0000;
    i=0;

    //Aktuator Stepper_Full Step
    if (freq<=5.5) // Valve membuka
    {
        berotasi (0,0,0,1); //step1
        berotasi (0,0,1,0); //step2
        berotasi (0,1,0,0); //step3
        berotasi (1,0,0,0); //step4
    }

    else if (freq>=5.5) // Valve Menutup
    {
        //Half Step
        tampil_LCD();
        berotasi (1,0,0,0); //step1
        berotasi (0,1,0,0); //step2
        berotasi (0,0,1,0); //step3
        berotasi (0,0,0,1); //step4
    }

    else if (5.2>freq<5.8) // valve berhenti
    {
        berotasi (0,0,0,0);
    }
}

```

```

    }
}

```

```

void layli ()

```

```

{
    startup();
    cond_1();
    cond_2();
    cond_3();
    cond_4();
    cond_5();
    cond_6();
    cond_7();

```

```

    lcd_clear();

```

```

    /*****

```

```

        MPX5500DP => Sensor 1 pada tangki 1

```

```

    *****/

```

```

        x=read_adc(0);

```

```

        c1=b-xmin;

```

```

        d1=a*c1;

```

```

        f1=e-d1;

```

```

        tangki1=((a/c1)*x)-(f1/c1);    //Pembacaan preesure pada

```

```

        sensor 1 => Psi

```

```

        h1=((float)(x*5)/255);

```

```

    /*****

```

```

        MPX5500DP => Sensor 2 pada tangki 2

```

```

    *****/

```

```

        y=read_adc(1);

```

```

        c2=b-ymin;

```

```

        d2=a*c2;

```

```

        f2=e-d2;

```

```

    tangki2=((a/c2)*y)-(f2/c2);    //Pembacaan pressure pada
    sensor 2 => Psi
    h2=((float)(y*5)/255);

    //*****
    *****

}

void wahyu ()
{
    //sensor H2s Kontrol
    sensor3 = read_adc(2);
    //  VinH2S1= ((sensor3)*5)/255;
    //  RO0 =4.38;
    //  RS0 = ((5-VinH2S1)/VinH2S1)*10;
    //  R0=RS0/RO0;
    //  pangkat0= 1/(R0*R0*R0); //R0 di pangkat -3
    //  ppm0 = 40*pangkat0;
    hs_pangkat = sensor3;

}

//Fungsi Deklarasi Port Yang digunakan  menggerakkan
//Stepper
void berotasi(unsigned char b3, unsigned char b2,unsigned
char b1, unsigned char b0)
{
    PORTA.0 = b3;
    PORTA.2 = b2;
    PORTA.4 = b1;
    PORTA.6 = b0;
    delay_ms (5);
}

void tampil_LCD()

```



```

{
    //-----Out LCD Real Time Clock-----
    -----//
    if (PINE.5==0)
    {
        lcd_clear();
        lcd_gotoxy(0,0);
        lcd_puts("MONITORING GAS (Ppm)");

        lcd_gotoxy(0,1);
        lcd_puts("CH4:");
        lcd_puts(tampil1);

        lcd_gotoxy(10,1);
        lcd_puts("CH4:");
        lcd_puts(tampil2);

        lcd_gotoxy(0,2);
        lcd_puts("CO2:");
        lcd_puts(tampil3);

        lcd_gotoxy(10,2);
        lcd_puts("CO2:");
        lcd_puts(tampil4);

        lcd_gotoxy(0,3);
        lcd_puts("H2S:");
        sprintf(buf6,"%f",hsl_pangkat2);
        lcd_puts(buf6);
        lcd_gotoxy(10,3);
        lcd_puts("H2S:");
        sprintf(buf7,"%f",hsl_pangkat3);
        lcd_puts(buf7);
    }
}

```

```
//-----Out LCD Monitoring-----  
-----//
```

```
else if (PINE.6 == 0)  
{
```

```
    lcd_clear();  
    lcd_gotoxy(0,0);  
    lcd_puts(" PURIFIKASI BIOGAS ");  
    lcd_gotoxy(0,1);  
    lcd_puts("-----");  
    //jam dan menit  
    rtc_get_time(&h,&m,&s);  
    rtc_get_date(0,&dd,&mm,&yy);  
    sprintf(buf0,"Jam : %02d:%02d:%02d",h,m,s);  
    lcd_gotoxy(0,2);  
    lcd_puts(buf0);
```

```
    //tanggal dan bulan  
    rtc_get_date(0,&dd,&mm,&yy);  
    sprintf(buf1,"Date:          %02d:%02d:%02d",dd,mm,yy);  
    //+2000);  
    lcd_gotoxy(0,3);  
    lcd_puts(buf1);  
}
```

```
//-----Out LCD Kontrol-----  
-----//
```

```
else if (PINE.7==0)  
{  
    lcd_clear();  
    lcd_gotoxy(0,0);  
    lcd_puts("Flow  =");  
    lcd_gotoxy(11,0);
```

```

        ftoa(freq,2,buf2);
        lcd_puts(buf2);
        lcd_gotoxy(17,0);
        lcd_putsf("L/m");
    lcd_gotoxy(0,1);
    lcd_puts("H2S =");
    //      sprintf(buf2,"%f",tangki1);
    //
    //      // sprintf(buf5,"%f", VinH2S1);
    //      lcd_gotoxy(11,1);
    //      lcd_puts(buf2);
    //      lcd_gotoxy(17,1);
        lcd_puts("Ppm");
    lcd_gotoxy(0,2);
    //lcd_puts("Pressure1=");
        sprintf(buf4,"Pressure1= %.2f",tangki1);
        lcd_gotoxy(11,2);
        lcd_puts(buf4);
        lcd_gotoxy(17,2);
        lcd_puts("Psi");
    lcd_gotoxy(0,3);
    lcd_puts("Pressure2=");
        sprintf(buf3,"%f",tangki2);
        lcd_gotoxy(11,3);
        lcd_puts(buf3);
        lcd_gotoxy(17,3);
        lcd_puts("Psi");
    }

    else
    {
    lcd_clear();
    lcd_gotoxy(0,0);
    lcd_puts(" SELAMAT DATANG");
    lcd_gotoxy(0,2);

```

```

        lcd_puts("Di Departemen");
        lcd_gotoxy(0,3);
        lcd_puts("Teknik Instrumentasi");
    }
}

void date()
{
    rtc_get_time(&h,&m,&s);
    rtc_get_date(0,&dd,&mm,&yy);
}

void startup()
{
    if (tangki1 <= 5 && tangki2 <= 5)
    {
        PORTA.5=1; //SV 1 => inputan tangki 1
        PORTA.3=0; //SV 2 => inputan tangki 2
        PORTA.7=0; //SV 3 => outputan tangki 1
        PORTA.1=0; //SV 4 => outputan tangki 2
    }
}

void cond_1()
{
    if (tangki1 >= 25 && tangki2 <= 5)
    {
        PORTA.5=0;
        PORTA.3=1;
        PORTA.7=1;
        PORTA.1=0;
        i=1;
    }
}

```

```

void cond_2()
{
if ((tangki1 <= 5 && (tangki2 <= 25 || tangki2 >= 25) &&
    l==1))
    {
        PORTA.5=0;
        PORTA.3=1;
        PORTA.7=0;
        PORTA.1=0;
        j=1;
    }
}
void cond_3()
{
if ((tangki1 <= 5 && tangki2 >=25) && j==1)
    {
        PORTA.5=1;
        PORTA.3=0;
        PORTA.7=0;
        PORTA.1=1;
        l=0;
        k=1;
    }
}
void cond_4()
{
if ((tangki1 >=5 && tangki2 <=5) && j==1 && l==0)
    {
        PORTA.5=1;
        PORTA.3=0;
        PORTA.7=0;
        PORTA.1=0;
    }
}
void cond_5()

```

```

{
if ((tangki1 >= 25 && tangki2 <=5))
{
    PORTA.5=0;
    PORTA.3=1;
    PORTA.7=1;
    PORTA.1=0;
    j=0;
    i=1;
}
}
void cond_6()
{
if (tangki1 >= 25 && (tangki2 <25 && tangki2 >5) && j==1)
{
    PORTA.5=0;
    PORTA.3=0;
    PORTA.7=0;
    PORTA.1=1;
}
}
void cond_7()
{
if ((tangki1 <25 && tangki1 >5) && (tangki2 >= 25) &&
    k==1)
{
    PORTA.5=0;
    PORTA.3=0;
    PORTA.7=1;
    PORTA.1=0;
    j=1;
}
}

void cond_8() //LOOPING WAHYU

```

```
{  
if (hs_pangkat>=3)  
{  
    PORTB.4=1;  
    PORTB.2=0;  
    PORTB.0=1;  
    PORTA.3=0;  
  
}  
}
```

LAMPIRAN B

(DATA SHEET SYSTEM ATMEGA128)

Features

- High-performance, Low-power Atmel® AVR® 8-bit Microcontroller
- Advanced RISC Architecture
 - 133 Powerful Instructions – Most Single Clock Cycle Execution
 - 32 x 8 General Purpose Working Registers + Peripheral Control Registers
 - Fully Static Operation
 - Up to 16MIPS Throughput at 16MHz
 - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory segments
 - 128Kbytes of In-System Self-programmable Flash program memory
 - 4Kbytes EEPROM
 - 4Kbytes Internal SRAM
 - Write/Erase cycles: 10,000 Flash/100,000 EEPROM
 - Data retention: 20 years at 85°C/100 years at 25°C⁽¹⁾
 - Optional Boot Code Section with Independent Lock Bits
 - In-System Programming by On-chip Boot Program
 - True Read-While-Write Operation
 - Up to 64Kbytes Optional External Memory Space
 - Programming Lock for Software Security
 - SPI Interface for In-System Programming
- QTouch® library support
 - Capacitive touch buttons, sliders and wheels
 - QTouch and QMatrix acquisition
 - Up to 64 sense channels
- JTAG (IEEE std. 1149.1 Compliant) Interface
 - Boundary-scan Capabilities According to the JTAG Standard
 - Extensive On-chip Debug Support
 - Programming of Flash, EEPROM, Fuses and Lock Bits through the JTAG Interface
- Peripheral Features
 - Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
 - Two Expanded 16-bit Timer/Counters with Separate Prescaler, Compare Mode and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Two 8-bit PWM Channels
 - 6 PWM Channels with Programmable Resolution from 2 to 16 Bits
 - Output Compare Modulator
 - 8-channel, 10-bit ADC
 - 8 Single-ended Channels
 - 7 Differential Channels
 - 2 Differential Channels with Programmable Gain at 1x, 10x, or 200x
 - Byte-oriented Two-wire Serial Interface
 - Dual Programmable Serial USARTs
 - Master/Slave SPI Serial Interface
 - Programmable Watchdog Timer with On-chip Oscillator
 - On-chip Analog Comparator
- Special Microcontroller Features
 - Power-on Reset and Programmable Brown-out Detection
 - Internal Calibrated RC Oscillator
 - External and Internal Interrupt Sources
 - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
 - Software Selectable Clock Frequency
 - ATmega103 Compatibility Mode Selected by a Fuse
 - Global Pull-up Disable
- I/O and Packages
 - 53 Programmable I/O Lines
 - 64-lead TQFP and 64-pad QFN/MLF
- Operating Voltages
 - 2.7 - 5.5V ATmega128L
 - 4.5 - 5.5V ATmega128
- Speed Grades
 - 0 - 8MHz ATmega128L
 - 0 - 16MHz ATmega128



**8-bit Atmel
Microcontroller
with 128KBytes
In-System
Programmable
Flash**

**ATmega128
ATmega128L**

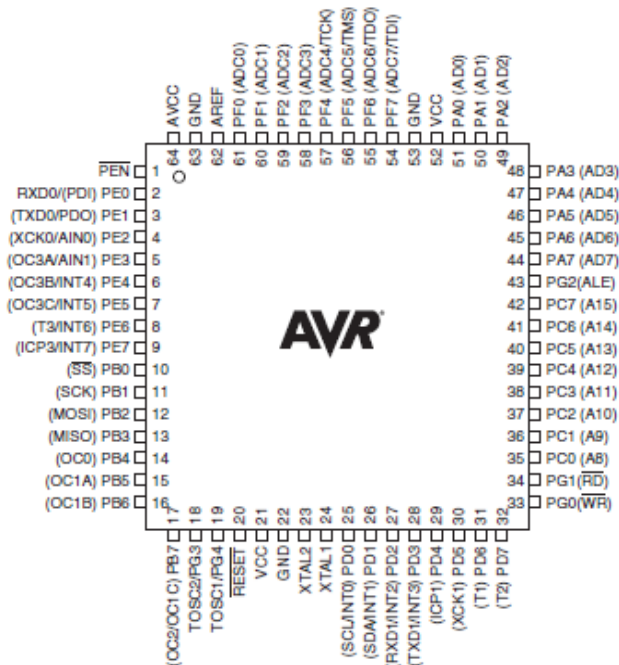
Summary

Rev. 2407X8-AV11-0011



Pin Configurations

Figure 1. Pinout ATmega128



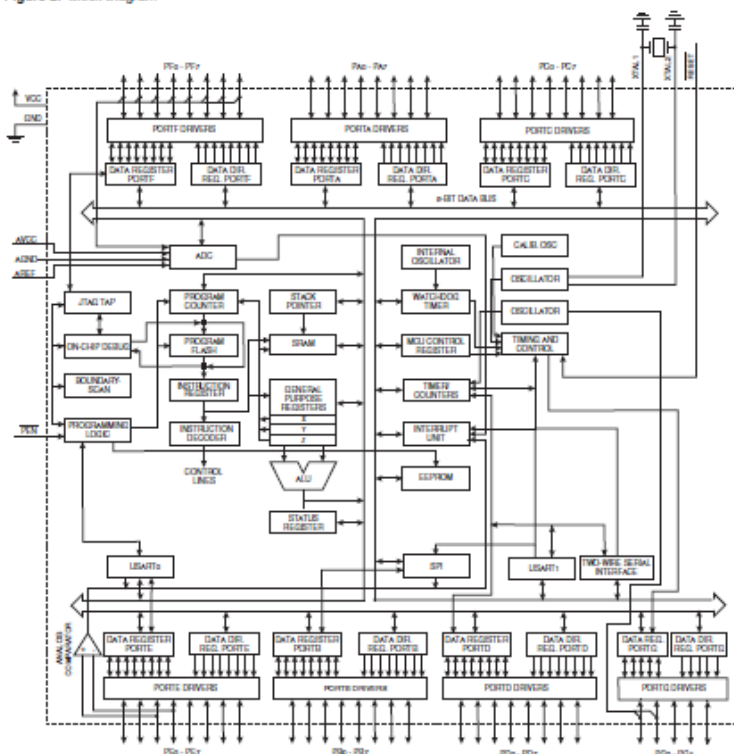
Note: The Pinout figure applies to both TQFP and MLF packages. The bottom pad under the QFN/MLF package should be soldered to ground.

Overview

The Atmel® AVR® ATmega128 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega128 achieves throughputs approaching 1MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

Block Diagram

Figure 2. Block Diagram



The Atmel® AVR® core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The ATmega128 provides the following features: 128Kbytes of In-System Programmable Flash with Read-While-Write capabilities, 4Kbytes EEPROM, 4Kbytes SRAM, 53 general purpose I/O lines, 32 general purpose working registers, Real Time Counter (RTC), four flexible Timer/Counters with compare modes and PWM, 2 USARTs, a byte oriented Two-wire Serial Interface, an 8-channel, 10-bit ADC with optional differential input stage with programmable gain, programmable Watchdog Timer with Internal Oscillator, an SPI serial port, IEEE std. 1149.1 compliant JTAG test interface, also used for accessing the On-chip Debug system and programming and six software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next interrupt or Hardware Reset. In Power-save mode, the asynchronous timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except Asynchronous Timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the Crystal/Resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low power consumption. In Extended Standby mode, both the main Oscillator and the Asynchronous Timer continue to run.

Atmel offers the QTouch® library for embedding capacitive touch buttons, sliders and wheels functionality into AVR microcontrollers. The patented charge-transfer signal acquisition offers robust sensing and includes fully debounced reporting of touch keys and includes Adjacent Key Suppression® (AKS™) technology for unambiguous detection of key events. The easy-to-use QTouch Suite toolchain allows you to explore, develop and debug your own touch applications.

The device is manufactured using Atmel's high-density nonvolatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed in-system through an SPI serial interface, by a conventional nonvolatile memory programmer, or by an On-chip Boot program running on the AVR core. The boot program can use any interface to download the application program in the application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega128 is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The ATmega128 device is supported with a full suite of program and system development tools including: C compilers, macro assemblers, program debugger/simulators, in-circuit emulators, and evaluation kits.

ATmega103 and ATmega128 Compatibility

The ATmega128 is a highly complex microcontroller where the number of I/O locations supersedes the 64 I/O locations reserved in the AVR instruction set. To ensure backward compatibility with the ATmega103, all I/O locations present in ATmega103 have the same location in ATmega128. Most additional I/O locations are added in an Extended I/O space starting from \$60 to \$FF, (i.e., in the ATmega103 internal RAM space). These locations can be reached by using LD/LDS/LDD and ST/STS/STD instructions only, not by using IN and OUT instructions. The relocation of the internal RAM space may still be a problem for ATmega103 users. Also, the increased number of interrupt vectors might be a problem if the code uses absolute addresses. To solve these problems, an ATmega103 compatibility mode can be selected by programming the fuse M103C. In this mode, none of the functions in the Extended I/O space are in use, so the internal RAM is located as in ATmega103. Also, the Extended Interrupt vectors are removed.

The ATmega128 is 100% pin compatible with ATmega103, and can replace the ATmega103 on current Printed Circuit Boards. The application note "Replacing ATmega103 by ATmega128" describes what the user should be aware of replacing the ATmega103 by an ATmega128.

ATmega103 Compatibility Mode

By programming the M103C fuse, the Atmel® ATmega128 will be compatible with the ATmega103 regards to RAM, I/O pins and interrupt vectors as described above. However, some new features in ATmega128 are not available in this compatibility mode, these features are listed below:

- One USART instead of two, Asynchronous mode only. Only the eight least significant bits of the Baud Rate Register is available.
- One 16 bits Timer/Counter with two compare registers instead of two 16-bit Timer/Counters with three compare registers.
- Two-wire serial interface is not supported.
- Port C is output only.
- Port G serves alternate functions only (not a general I/O port).
- Port F serves as digital input only in addition to analog input to the ADC.
- Boot Loader capabilities is not supported.
- It is not possible to adjust the frequency of the internal calibrated RC Oscillator.
- The External Memory Interface can not release any Address pins for general I/O, neither configure different wait-states to different External Memory Address sections.

In addition, there are some other minor differences to make it more compatible to ATmega103:

- Only EXTRF and PORF exists in MCUCSR.
- Timed sequence not required for Watchdog Time-out change.
- External Interrupt pins 3 - 0 serve as level interrupt only.
- USART has no FIFO buffer, so data overrun comes earlier.

Unused I/O bits in ATmega103 should be written to 0 to ensure same operation in ATmega128.

Pin Descriptions

VCC Digital supply voltage.

GND Ground.

Port A (PA7..PA0) Port A is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port A output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port A pins that are externally pulled low will source current if the pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port A also serves the functions of various special features of the ATmega128 as listed on [page 72](#).

Port B (PB7..PB0) Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port B also serves the functions of various special features of the ATmega128 as listed on [page 73](#).

Port C (PC7..PC0)	<p>Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p> <p>Port C also serves the functions of special features of the Atmel® AVR® ATmega128 as listed on page 76. In ATmega103 compatibility mode, Port C is output only, and the port C pins are not tri-stated when a reset condition becomes active.</p> <p>Note: The ATmega128 is by default shipped in ATmega103 compatibility mode. Thus, if the parts are not programmed before they are put on the PCB, PORTC will be output during first power up, and until the ATmega103 compatibility mode is disabled.</p>
Port D (PD7..PD0)	<p>Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p> <p>Port D also serves the functions of various special features of the ATmega128 as listed on page 77.</p>
Port E (PE7..PE0)	<p>Port E is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port E output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port E pins that are externally pulled low will source current if the pull-up resistors are activated. The Port E pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p> <p>Port E also serves the functions of various special features of the ATmega128 as listed on page 80.</p>
Port F (PF7..PF0)	<p>Port F serves as the analog inputs to the A/D Converter.</p> <p>Port F also serves as an 8-bit bi-directional I/O port, if the A/D Converter is not used. Port pins can provide internal pull-up resistors (selected for each bit). The Port F output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port F pins that are externally pulled low will source current if the pull-up resistors are activated. The Port F pins are tri-stated when a reset condition becomes active, even if the clock is not running. If the JTAG interface is enabled, the pull-up resistors on pins PF7(TDI), PF5(TMS), and PF4(TCK) will be activated even if a Reset occurs.</p> <p>The TDO pin is tri-stated unless TAP states that shift out data are entered.</p> <p>Port F also serves the functions of the JTAG interface.</p> <p>In ATmega103 compatibility mode, Port F is an input Port only.</p>
Port G (PG4..PG0)	<p>Port G is a 5-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port G output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port G pins that are externally pulled low will source current if the pull-up resistors are activated. The Port G pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p> <p>Port G also serves the functions of various special features.</p> <p>The Port G pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p>

In ATmega103 compatibility mode, these pins only serve as strobe signals to the external memory as well as input to the 32kHz Oscillator, and the pins are initialized to PG0 = 1, PG1 = 1, and PG2 = 0 asynchronously when a reset condition becomes active, even if the clock is not running. PG3 and PG4 are oscillator pins.

RESET	Reset input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The minimum pulse length is given in Table 19 on page 50 . Shorter pulses are not guaranteed to generate a reset.
XTAL1	Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.
XTAL2	Output from the inverting Oscillator amplifier.
AVCC	AVCC is the supply voltage pin for Port F and the A/D Converter. It should be externally connected to V_{CC} , even if the ADC is not used. If the ADC is used, it should be connected to V_{CC} through a low-pass filter.
AREF	AREF is the analog reference pin for the A/D Converter.
PEN	PEN is a programming enable pin for the SPI Serial Programming mode, and is internally pulled high. By holding this pin low during a Power-on Reset, the device will enter the SPI Serial Programming mode. PEN has no function during normal operation.

Resources

A comprehensive set of development tools, application notes, and datasheets are available for download on <http://www.atmel.com/avr>.

Data Retention

Reliability Qualification results show that the projected data retention failure rate is much less than 1 PPM over 20 years at 85°C or 100 years at 25°C

About Code Examples

This datasheet contains simple code examples that briefly show how to use various parts of the device. These code examples assume that the part specific header file is included before compilation. Be aware that not all C compiler vendors include bit definitions in the header files and interrupt handling in C is compiler dependent. Please confirm with the C compiler documentation for more details.

For I/O registers located in extended I/O map, "IN", "OUT", "SBIS", "SBIC", "CBI", and "SBI" instructions must be replaced with instructions that allow access to extended I/O. Typically "LDS" and "STS" combined with "SBR", "SBR", "SBR", and "CBR".

Capacitive touch sensing

The Atmel QTouch Library provides a simple to use solution to realize touch sensitive interfaces on most Atmel AVR microcontrollers. The QTouch Library includes support for the QTouch and QMatrix acquisition methods.

Touch sensing can be added to any application by linking the appropriate Atmel QTouch Library for the AVR Microcontroller. This is done by using a simple set of APIs to define the touch channels and sensors, and then calling the touch sensing API's to retrieve the channel information and determine the touch sensor states.

The QTouch Library is FREE and downloadable from the Atmel website at the following location: www.atmel.com/qtouchlibrary. For implementation details and other information, refer to the [Atmel QTouch Library User Guide](#) - also available for download from the Atmel website.

Instruction Set Summary

Mnemonic	Operands	Description	Operation	Flags	#Cycles
ARITHMETIC AND LOGIC INSTRUCTIONS					
ADD	rd, r	Add Two Registers	$rd \leftarrow rd + r$	Z, C, N, V, H	1
ADC	rd, r	Add with Carry Two Registers	$rd \leftarrow rd + r + C$	Z, C, N, V, H	1
ADIW	rd, K	Add Immediate to Word	$rd \leftarrow rd + K$	Z, C, N, V, H	2
SUB	rd, r	Subtract Two Registers	$rd \leftarrow rd - r$	Z, C, N, V, H	1
SLD	rd, K	Subtract Constant from Register	$rd \leftarrow rd - K$	Z, C, N, V, H	1
SEC	rd, r	Subtract with Carry Two Registers	$rd \leftarrow rd - r - C$	Z, C, N, V, H	1
SED	rd, K	Subtract with Carry Constant from Register	$rd \leftarrow rd - K - C$	Z, C, N, V, H	1
SRW	rd, K	Subtract Immediate from Word	$rd \leftarrow rd - K$	Z, C, N, V, H	2
AND	rd, r	Logical AND Registers	$rd \leftarrow rd \wedge r$	Z, N, V	1
ANDI	rd, K	Logical AND Register and Constant	$rd \leftarrow rd \wedge K$	Z, N, V	1
OR	rd, r	Logical OR Registers	$rd \leftarrow rd \vee r$	Z, N, V	1
ORI	rd, K	Logical OR Register and Constant	$rd \leftarrow rd \vee K$	Z, N, V	1
LDOR	rd, r	Exclusive OR Registers	$rd \leftarrow rd \oplus r$	Z, N, V	1
COM	rd	One's Complement	$rd \leftarrow \sim rd$	Z, C, N, V	1
NEG	rd	Two's Complement	$rd \leftarrow -rd$	Z, C, N, V, H	1
SEI	SREG	Set Status in Register	$SREG \leftarrow 0xFF$	Z, N, V	1
CLI	SREG	Clear Status in Register	$SREG \leftarrow 0x00$	Z, N, V	1
INC	rd	Increment	$rd \leftarrow rd + 1$	Z, N, V	1
DEC	rd	Decrement	$rd \leftarrow rd - 1$	Z, N, V	1
TSR	rd	Test for Zero or Minus	$rd \leftarrow rd \wedge rd$	Z, N, V	1
CLR	rd	Clear Register	$rd \leftarrow 0x00$	Z, N, V	1
SEI	SREG	Set Register	$SREG \leftarrow 0xFF$	None	1
MUL	rd, r	Multiply Unsigned	$rd \leftarrow rd \times r$	Z, C	2
MULS	rd, r	Multiply Signed	$rd \leftarrow rd \times r$	Z, C	2
MULSU	rd, r	Multiply Signed with Unsigned	$rd \leftarrow rd \times r$	Z, C	2
FMULS	rd, r	Fractional Multiply Signed	$rd \leftarrow (rd \times r) \ll 1$	Z, C	2
FMULSU	rd, r	Fractional Multiply Signed with Unsigned	$rd \leftarrow (rd \times r) \ll 1$	Z, C	2
BRANCH INSTRUCTIONS					
JMP	k	Relative Jump	$PC \leftarrow PC + k + 1$	None	2
JMP	k	Indirect Jump to ZP	$PC \leftarrow ZP$	None	2
JMP	k	Direct Jump	$PC \leftarrow k$	None	3
RCALL	k	Relative Subroutine Call	$PC \leftarrow PC + k + 1$	None	3
CALL	k	Indirect Call to ZP	$PC \leftarrow ZP$	None	3
CALL	k	Direct Subroutine Call	$PC \leftarrow k$	None	4
RET		Subroutine Return	$PC \leftarrow STACK$	None	4
RETI		Interrupt Return	$PC \leftarrow STACK$	1	4
CPSE	rd, r	Compare, Skip if Equal	$if (rd == r) PC \leftarrow PC + 2 or 3$	None	1/2/3
CP	rd, r	Compare	$rd - r$	Z, N, V, C, H	1
CPSE	rd, r	Compare with Carry	$rd - r - C$	Z, N, V, C, H	1
CP	rd, r	Compare Register with Immediate	$rd - K$	Z, N, V, C, H	1
SENC	rd, r	Skip if Set in Register Cleared	$if (rd \wedge r) PC \leftarrow PC + 2 or 3$	None	1/2/3
SESR	rd, r	Skip if Set in Register is Set	$if (rd \wedge r) PC \leftarrow PC + 2 or 3$	None	1/2/3
SENC	r, r	Skip if Set in ZC Register Cleared	$if (rd \wedge r) PC \leftarrow PC + 2 or 3$	None	1/2/3
SESR	r, r	Skip if Set in ZC Register is Set	$if (rd \wedge r) PC \leftarrow PC + 2 or 3$	None	1/2/3
SESR	k, k	Branch if Status Flag Set	$if (SREG \wedge k) PC \leftarrow PC + k + 1$	None	1/2
SESR	k, k	Branch if Status Flag Cleared	$if (SREG \wedge k) PC \leftarrow PC + k + 1$	None	1/2
SESR	k	Branch if Equal	$if (Z \wedge k) PC \leftarrow PC + k + 1$	None	1/2
SESR	k	Branch if Not Equal	$if (\sim Z \wedge k) PC \leftarrow PC + k + 1$	None	1/2
SESR	k	Branch if Carry Set	$if (C \wedge k) PC \leftarrow PC + k + 1$	None	1/2
SESR	k	Branch if Carry Cleared	$if (\sim C \wedge k) PC \leftarrow PC + k + 1$	None	1/2
SESR	k	Branch if Same or Higher	$if (C \wedge k) PC \leftarrow PC + k + 1$	None	1/2
SESR	k	Branch if Lower	$if (\sim C \wedge k) PC \leftarrow PC + k + 1$	None	1/2
SESR	k	Branch if Minus	$if (N \wedge k) PC \leftarrow PC + k + 1$	None	1/2
SESR	k	Branch if Plus	$if (\sim N \wedge k) PC \leftarrow PC + k + 1$	None	1/2
SESR	k	Branch if Greater or Equal, Signed	$if (N \wedge V, k) PC \leftarrow PC + k + 1$	None	1/2
SESR	k	Branch if Less Than Zero, Signed	$if (\sim N \wedge V, k) PC \leftarrow PC + k + 1$	None	1/2
SESR	k	Branch if Carry Flag Set	$if (C \wedge k) PC \leftarrow PC + k + 1$	None	1/2
SESR	k	Branch if Carry Flag Cleared	$if (\sim C \wedge k) PC \leftarrow PC + k + 1$	None	1/2
SESR	k	Branch if T Flag Set	$if (T \wedge k) PC \leftarrow PC + k + 1$	None	1/2
SESR	k	Branch if T Flag Cleared	$if (\sim T \wedge k) PC \leftarrow PC + k + 1$	None	1/2
SESR	k	Branch if Overflow Flag is Set	$if (O \wedge k) PC \leftarrow PC + k + 1$	None	1/2
SESR	k	Branch if Overflow Flag is Cleared	$if (\sim O \wedge k) PC \leftarrow PC + k + 1$	None	1/2

Instruction Set Summary (Continued)

Mnemonic	Operands	Description	Operation	Flags	#Cycles
SEI	s	Branch if Interrupt Enabled	$Z \leftarrow (1 - I) \text{ then } PC \leftarrow PC + s + 1$	None	1/2
CLI	s	Branch if Interrupt Disabled	$Z \leftarrow (1 - I) \text{ then } PC \leftarrow PC + s + 1$	None	1/2
DATA TRANSFER INSTRUCTIONS					
MOV	Rd, Rr	Move Between Registers	$Rd \leftarrow Rr$	None	1
MOVW	Rd, Rr	Copy Register Word	$Rd \leftarrow Rr \ll 16 \vee Rr \gg 16$	None	1
LDI	Rd, K	Load Immediate	$Rd \leftarrow K$	None	1
LDD	Rd, Rr	Load Indirect	$Rd \leftarrow [Rr]$	None	2
LDD	Rd, Rr	Load Indirect and Post-Inc.	$Rd \leftarrow [Rr]; Rr \leftarrow Rr + 1$	None	2
LDD	Rd, Rr	Load Indirect and Pre-Dec.	$Rr \leftarrow Rr - 1; Rd \leftarrow [Rr]$	None	2
LDD	Rd, Y	Load Indirect	$Rd \leftarrow [Y]$	None	2
LDD	Rd, Y+	Load Indirect and Post-Inc.	$Rd \leftarrow [Y]; Y \leftarrow Y + 1$	None	2
LDD	Rd, Y-	Load Indirect and Pre-Dec.	$Y \leftarrow Y - 1; Rd \leftarrow [Y]$	None	2
LDD	Rd, Y+Lo	Load Indirect with Displacement	$Rd \leftarrow [Y + Lo]$	None	2
LDD	Rd, Z	Load Indirect	$Rd \leftarrow [Z]$	None	2
LDD	Rd, Z+	Load Indirect and Post-Inc.	$Rd \leftarrow [Z]; Z \leftarrow Z + 1$	None	2
LDD	Rd, Z-	Load Indirect and Pre-Dec.	$Z \leftarrow Z - 1; Rd \leftarrow [Z]$	None	2
LDD	Rd, Z+Lo	Load Indirect with Displacement	$Rd \leftarrow [Z + Lo]$	None	2
LDS	Rd, K	Load Direct from SRAM	$Rd \leftarrow [K]$	None	2
ST	X, Rr	Store Indirect	$[X] \leftarrow Rr$	None	2
ST	X+, Rr	Store Indirect and Post-Inc.	$[X] \leftarrow Rr; X \leftarrow X + 1$	None	2
ST	X-, Rr	Store Indirect and Pre-Dec.	$X \leftarrow X - 1; [X] \leftarrow Rr$	None	2
ST	Y, Rr	Store Indirect	$[Y] \leftarrow Rr$	None	2
ST	Y+, Rr	Store Indirect and Post-Inc.	$[Y] \leftarrow Rr; Y \leftarrow Y + 1$	None	2
ST	Y-, Rr	Store Indirect and Pre-Dec.	$Y \leftarrow Y - 1; [Y] \leftarrow Rr$	None	2
STD	Y+Lo, Rr	Store Indirect with Displacement	$[Y + Lo] \leftarrow Rr$	None	2
ST	Z, Rr	Store Indirect	$[Z] \leftarrow Rr$	None	2
ST	Z+, Rr	Store Indirect and Post-Inc.	$[Z] \leftarrow Rr; Z \leftarrow Z + 1$	None	2
ST	Z-, Rr	Store Indirect and Pre-Dec.	$Z \leftarrow Z - 1; [Z] \leftarrow Rr$	None	2
STD	Z+Lo, Rr	Store Indirect with Displacement	$[Z + Lo] \leftarrow Rr$	None	2
STS	K, Rr	Store Direct to SRAM	$[K] \leftarrow Rr$	None	2
LPM	Rd, Z	Load Program Memory	$Rd \leftarrow [Z]$	None	3
LPM	Rd, Z	Load Program Memory	$Rd \leftarrow [Z]$	None	3
LPM	Rd, Z+	Load Program Memory and Post-Inc.	$Rd \leftarrow [Z]; Z \leftarrow Z + 1$	None	3
ELPM	Rd, Z	Extended Load Program Memory	$Rd \leftarrow [RAMFZ]; Z \leftarrow Z + 1$	None	3
ELPM	Rd, Z	Extended Load Program Memory	$Rd \leftarrow [RAMFZ]; Z \leftarrow Z + 1$	None	3
ELPM	Rd, Z+	Extended Load Program Memory and Post-Inc.	$Rd \leftarrow [RAMFZ]; Z \leftarrow Z + 1; Z \leftarrow Z + 1$	None	3
SPM		Store Program Memory	$[Z] \leftarrow [R0]$	None	-
IN	Rd, P	In Port	$Rd \leftarrow P$	None	1
OUT	P, Rr	Out Port	$P \leftarrow Rr$	None	1
PUSH	Rr	Push Register on Stack	$STACK \leftarrow Rr$	None	2
POP	Rd	Pop Register from Stack	$Rd \leftarrow STACK$	None	2
BIT AND BIT-TEST INSTRUCTIONS					
SB	P, b	Set Bit in I/O Register	$IOP[b] \leftarrow 1$	None	2
CB	P, b	Clear Bit in I/O Register	$IOP[b] \leftarrow 0$	None	2
LSB	Rd	Logical Shift Left	$REG[b+1] \leftarrow REG[b]; REG[b] \leftarrow 0$	Z, C, N, V	1
LSSR	Rd	Logical Shift Right	$REG[b] \leftarrow REG[b]; REG[b+1] \leftarrow 0$	Z, C, N, V	1
ROL	Rd	Rotate Left Through Carry	$REG[b+1] \leftarrow REG[b]; REG[b] \leftarrow REG[b] \oplus CF$	Z, C, N, V	1
ROR	Rd	Rotate Right Through Carry	$REG[b] \leftarrow REG[b] \oplus CF; REG[b+1] \leftarrow REG[b]$	Z, C, N, V	1
ASR	Rd	Arithmetic Shift Right	$REG[b+1] \leftarrow REG[b]; REG[b] \leftarrow REG[b] \gg 1$	Z, C, N, V	1
SWAP	Rd	Swap Nibbles	$REG[b+1] \leftarrow REG[b]; REG[b] \leftarrow REG[b+1]$	None	1
SEI	s	Flag Set	$SREG[SE] \leftarrow 1$	SREG[SE]	1
CLI	s	Flag Clear	$SREG[SE] \leftarrow 0$	SREG[SE]	1
ST	Rr, b	St Store from Register to T	$T \leftarrow Rr[b]$	T	1
LD	Rd, b	St Load from T to Register	$Rd[b] \leftarrow T$	None	1
SCD		Set Carry	$CF \leftarrow 1$	C	1
CSD		Clear Carry	$CF \leftarrow 0$	C	1
SEN		Set Negative Flag	$N \leftarrow 1$	N	1
CN		Clear Negative Flag	$N \leftarrow 0$	N	1
SEZ		Set Zero Flag	$Z \leftarrow 1$	Z	1
CZ		Clear Zero Flag	$Z \leftarrow 0$	Z	1
SET		Global Interrupt Enable	$I \leftarrow 1$	I	1
CLT		Global Interrupt Disable	$I \leftarrow 0$	I	1
SEB		Set Signed Test Flag	$S \leftarrow 1$	S	1
CLB		Clear Signed Test Flag	$S \leftarrow 0$	S	1

Instruction Set Summary (Continued)

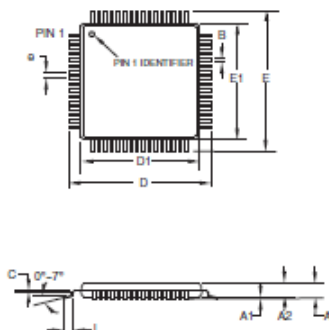
Mnemonic	Operands	Description	Operation	Flags	#Cycles
SEV		Set Zero Complement Overflow	$Z \leftarrow 1$	Z	1
CLV		Clear Zero Complement Overflow	$Z \leftarrow 0$	Z	1
SET		Set T in SREG	$T \leftarrow 1$	T	1
CLT		Clear T in SREG	$T \leftarrow 0$	T	1
SEH		Set Half-Carry Flag in SREG	$H \leftarrow 1$	H	1
CLH		Clear Half-Carry Flag in SREG	$H \leftarrow 0$	H	1
WDT CONTROL INSTRUCTIONS					
NOP		No Operation		None	1
SLEEP		Sleep	(see specific desc. for Sleep function)	None	1
WDR		Watchdog Reset	(see specific desc. for WDR function)	None	1
BREAK		Break	For On-chip Debug Only	None	N/A

Ordering Information

Speed (MHz)	Power Supply	Ordering Code ⁽¹⁾	Package ⁽²⁾	Operation Range
8	2.7 – 5.5V	ATmega128L-8AU ATmega128L-8AUR ⁽³⁾ ATmega128L-8MU ATmega128L-8MUR ⁽³⁾	64A 64A 64M1 64M1	Industrial (-40°C to 85°C)
16	4.5 – 5.5V	ATmega128-16AU ATmega128-16AUR ⁽³⁾ ATmega128-16MU ATmega128-16MUR ⁽³⁾	64A 64A 64M1 64M1	
8	3.0 – 5.5V	ATmega128L-8AN ATmega128L-8ANR ⁽³⁾ ATmega128L-8MN ATmega128L-8MNR ⁽³⁾	64A 64A 64M1 64M1	Extended (-40°C to 105°C)
16	4.5 – 5.5V	ATmega128-16AN ATmega128-16ANR ⁽³⁾ ATmega128-16MN ATmega128-16MNR ⁽³⁾	64A 64A 64M1 64M1	

- Notes: 1. Pb-free packaging complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.
 2. The device can also be supplied in water form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.
 3. Tape and Reel

Package Type	
64A	64-lead, 14 x 14 x 1.0mm, Thin Profile Plastic Quad Flat Package (TQFP)
64M1	64-pad, 9 x 9 x 1.0mm, Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)

Packaging Information
64A

COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	—	—	1.30	
A1	0.05	—	0.15	
A2	0.05	1.00	1.05	
D	15.75	16.00	16.25	
D1	13.90	14.00	14.10	Note 2
E	15.75	16.00	16.25	
E1	13.90	14.00	14.10	Note 2
B	0.30	—	0.45	
C	0.09	—	0.20	
L	0.45	—	0.75	
s	0.80 TYP			

Notes:

1. This package conforms to JEDEC reference MS-018, Variation AEB.
2. Dimensions D1 and E1 do not include mold protrusion. Allowable protrusion is 0.25 mm per side. Dimensions D1 and E1 are maximum plastic body size dimensions including mold mismatch.
3. Lead coplanarity is 0.10 mm maximum.

2010-10-20


2325 Orchard Parkway
San Jose, CA 95131

TITLE
64A, 64-lead, 14 x 14 mm Body Size, 1.0 mm Body Thickness,
0.8 mm Lead Pitch, Thin Profile Plastic Quad Flat Package (TQFP)

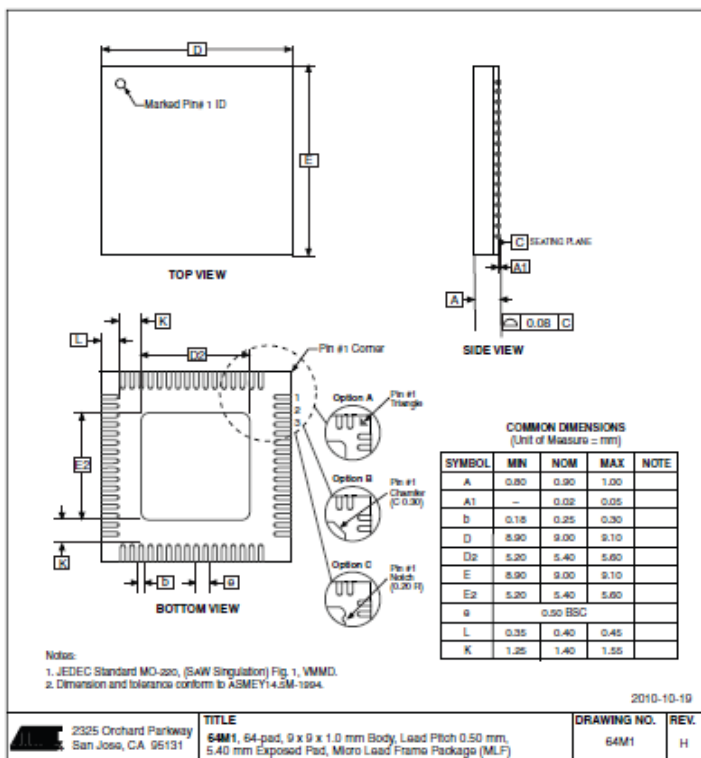
DRAWING NO.

64A

REV.

C

64M1



LAMPIRAN C
(DATA SHEET SENSOR FLOW TYPE FS400A)



Gambar C. Water flowmeter FS400A 1"

Spesifikasi G 1 inch water flow sensor :

- Rentang tegangan operasional antara 5 ~ 24 Volt DC
- Maksimum arus operasional sebesar 15 mA (pada 5VDC)
- Maksimum suhu air 120°C
- Maksimum suhu operasional 80°C
- Rentang kelembapan operasional antara 35% ~ 90% RH
- Maksimum tekanan air 1,2 MPa
- Diameter thread inlet/outlet 1 $\frac{1}{2}$ (~25 mm)
- Range flow : 1 liter/ menit ;V 60 liter/menit
- Calculation Formula : $F(Hz)=4.8*Q(L/min)$

Catatan:

- Ketika bahan magnetik dekat dengan sensor ,karakteristiknya dapat bervariasi .
- Untuk menghindari puing-puing partikel , sensor harus dipasang setelah filter .
- Aliran instalasi sensor harus menghindari getaran yang kuat dan gemetar lingkungan , sehingga tidak mempengaruhi akurasi pengukuran sensor .

LAMPIRAN D
(*MANUAL BOOK*)

BIOGAS PURIFICATION SYSTEM MANUAL BOOK



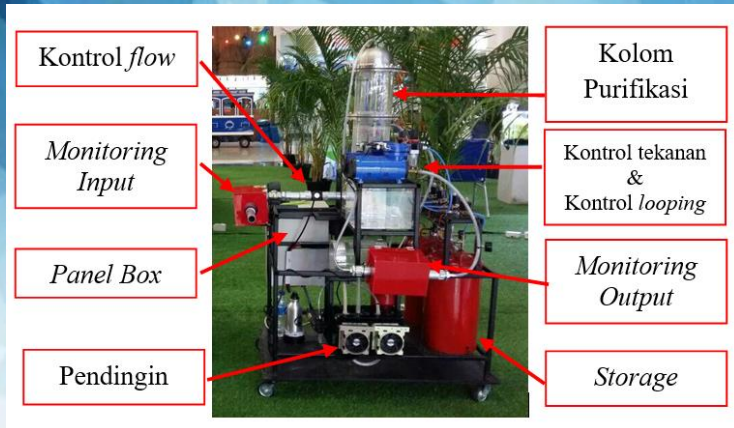
GAMBARAN UMUM ALAT

Biogas Purification System adalah gagasan inovasi dalam pemurnian biogas yang mengkombinasikan antara sistem absorpsi fisika dengan sistem adsorpsi kimia dalam satu kolom yang dirancang secara otomatis. Sistem absorpsi fisika yang digunakan pada alat ini menggunakan *water scrubber* dengan temperatur air antara 10-15°C. Hasil dari absorpsi fisika ini dapat melarutkan kadar H₂S, CO₂ dan PM dengan tingkat kelarutan gas sebanding dengan rendahnya temperatur air pada *water scrubber*. Sedangkan pada sistem adsorpsi kimia alat ini menggunakan zat kimia CaO apabila bereaksi dengan gas biogas dapat mengurangi kadar CO₂ dan CaCl₂ apabila bereaksi dengan gas biogas dapat mengurangi kadar air dalam biogas. Selain itu ***Biogas Purification System*** juga dilengkapi dengan sistem kontrol meliputi sistem kontrol temperatur pada *water cooling system*, sistem kontrol tekanan pada *biogas storage*, serta sistem *monitoring* kadar gas H₂S, CO₂, CH₄ pada *inlet* dan *outlet* purifikasi biogas yang saling terintegrasi. ***Biogas Purification System*** sangat sesuai digunakan oleh pengguna biogas di Indonesia karena alat ini merupakan alat purifikasi yang efektif, efisien, dan sekaligus menggunakan bahan pemurnian yang relatif mudah untuk diregenerasi secara lokal di wilayah Indonesia.

SPESIFIKASI ALAT

1. Sistem Pendingin Air
2. Sensor Gas
 - H_2S : MQ-136
 - CH_4 : MQ-4
 - CO_2 : MG-811
3. Sensor Tekanan (MPX5500DP)
4. Mini Kompresor
5. Sensor Flow G1
6. *Water Scrubber System*
7. Mikrokontroler ATmega128
8. *Storage Tank*
 - Tinggi : 38 cm
 - Diameter: 21 cm
9. *Display LCD 4x20*
10. Kolom Purifikasi
 - Tinggi : 80 cm
 - Diameter: 15 cm
11. *Spray atau Nozzle*
12. *Sieve Plate Tray Trap*
13. *Stepper*
14. *Solenoid Valve*
15. Sensor DHT

DESKRIPSI ALAT



Gambar Plant Purifikasi Biogas

Plant ini dilengkapi dengan kontrol flow gas inlet, monitoring gas, kontrol H_2S , kontrol tekanan dan juga IoT. Adapun penjelasannya sebagai berikut :

1. Kontrol Flow Inlet

Sistem *control flow* ini bertujuan untuk mengatur laju aliran yang akan memasuki kolom purifikasi agar hasil dari biogas dalam purifikasi dapat lebih efektif.

Berikut adalah komponen yang digunakan pada sistem control flow inlet pada purifikasi biogas :

- Sensor Flowmeter G 1"

Sensor ini berfungsi untuk mendeteksi aliran biogas yang melewati pipa inlet purifikasi biogas. Sensor ini diatur dengan setpoint sebesar 5 L/min.

- ATmega 128
ATmega 128 merupakan kontroler yang berfungsi untuk mengontrol semua sinyal yang diperintahkan oleh sensor untuk menggerakkan actuator.
- Motor Stepper
Motor stepper adalah actuator atau penggerak. Motor stepper ini juga bisa disebut keran otomatis yang dapat membuka atau menutup secara otomatis ketika laju aliran yang telah dideteksi oleh sensor tidak sesuai dengan set point yang telah ditetapkan yaitu 5 L/min. Motor stepper ini juga dilengkapi dengan driver motor yang berada pada panel box guna untuk pengkondisian sinyal yang akan diproses pada ATmega128.

2. Monitoring Gas

Sistem monitoring kadar gas ini meliputi gas CH₄, CO₂, dan H₂S. sistem monitoring ini berfungsi untuk tujuan sehingga dapat mengetahui kualitas kandungan gas CH₄, karena kadar CH₄ sangat menentukan kualitas dari biogas. Sedangkan kadar CO₂ dan H₂S perlu dimonitoring untuk memastikan keamanan, melindungi, mengantisipasi serta menyiapkan pemeliharaan / maintenance peralatan sebab kadung gas tersebut merupakan kadar pengotor pada biogas yang dapat merusak peralat pengaplikasian biogas.

Sistem monitoring ini terletak pada bagian inlet dan outlet kolom purifikasi, sehingga dapat terlihat perbandingan antara biogas sebelum atau sesudah

memasuki kolom purifikasi dan dapat terlihat efektifitas dari proses purifikasi tersebut.

Berikut ini adalah komponen yang digunakan untuk sistem monitoring kadar gas pada sistem purifikasi biogas :

- Sensor (MG-811, MQ-136 dan MQ-4)
Sensor ini berfungsi untuk mendeteksi kandungan gas yang berada pada biogas. Sensor MG-811 berfungsi untuk mendeteksi kadar CO₂. MQ-136 berfungsi untuk mendeteksi H₂S. dan sensor MQ-4 dapat berfungsi untuk mendeteksi gas CH₄.
- ATmega 128
ATmega 128 berfungsi sebagai prosesor yang memproses seluruh sinyal agar dapat bekerja sesuai yang diinginkan.
- RTC (Real Time Clock)
RTC ini berfungsi sebagai penampil waktu yang sebenarnya. Dilengkapi dengan baterai CMOS agar dapat selalu berkerja meskipun pant dalam kondisi mati, dan ketika dihidupkan maka waktu yang ditampilkan dapat menunjukan waktu sebenarnya tanpa harus mensetting ulang.
- SDcard
SDcard merupakan memory yang digunakan untuk merekam/menyimpan seluruh hasil variable yang telah diukur. SDcard ini mampu menyimpan data yang bekerja setiap detik.

3. Kontrol H₂S

Sistem *control* H₂S ini berfungsi untuk mengatur kadar H₂S yang akan keluar pada kolom purifikasi, agar kadar biogas yang keluar dari kolom purifikasi dapat memenuhi standar yang telah diinginkan.

Berikut adalah komponen yang digunakan pada sistem control H₂S :

- Sensor H₂S (MQ-136)
Sensor ini berfungsi untuk mendeteksi kadar H₂S yang ada pada biogas. Sensor ini diatur dengan setpoint sebesar 3 ppm.
- ATmega 128
ATmega 128 merupakan kontroler yang berfungsi untuk mengontrol semua sinyal yang diperintahkan oleh sensor untuk menggerakkan aktuator.
- Solenoid
Solenoid berfungsi sebagai aktuator atau penggerak. Pada sistem ini menggunakan 2 solenoid untuk mengatur aliran biogas. Solenoid (a) akan menyala ketika sensor mendeteksi kadar H₂S > 3 ppm. Yang artinya aliran biogas akan ditarik kembali untuk memasuki kolom purifikasi untuk di proses ulang. Sedangkan solenoid (b) akan aktif ketika kadar H₂S < 3 ppm. Maka aliran biogas akan diteruskan keluar dari kolom purifikasi menuju proses selanjutnya.

4. Kontrol Tekanan

Sistem kontrol tekanan berfungsi untuk mengatur tekanan dalam *storage tank* agar tekanan didalamnya bisa tetap stabil dan dapat sesuai dengan yang telah ditentukan.

Komponen yang digunakan pada system kontrol tekanan ini meliputi:

- Sensor tekanan (MPX5500DP)
Sensor tekanan difungsikan untuk mendeteksi tekanan didalam *storage tank* dimana *setpoint* yang digunakan sebesar 5Psi sebagai batas bawah dan 25Psi sebagai batas atas.
- Mikrokontroller ATmega 128
Mikrokontroller berfungsi sebagai pengontrol semua sinyal yang berasal dari sensor untuk menggerakkan actuator.
- *Solenoid Valve*
Solenoid Valve berfungsi sebagai aktuator yang bersifat *on/off*. *Solenoid valve* yang digunakan pada kontrol tekanan ini berjumlah 4 buah yaitu *solenoid valve* 1 dan 2 sebagai input kemudian *solenoid valve* 3 dan 4 sebagai output. Input akan membuka ketika tekanan 5 Psi dan menutup ketika tekanan telah mencapai 25 Psi, sedangkan *solenoid valve* 3 dan 4 akan membuka ketika tekanan 25 Psi dan menutup ketika tekanan 5 Psi. Untuk tahapan terakhir outputan gas dari *storage tank* akan diteruskan menuju generator.

5. IoT

Sistem kontrol tekanan berfungsi untuk mengatur tekanan dalam *storage tank* agar tekanan didalamnya bisa tetap stabil dan dapat sesuai dengan yang telah ditentukan.

Komponen yang digunakan pada system kontrol tekanan ini meliputi:

1. Raspberry Pi3



2. Smartphone Android

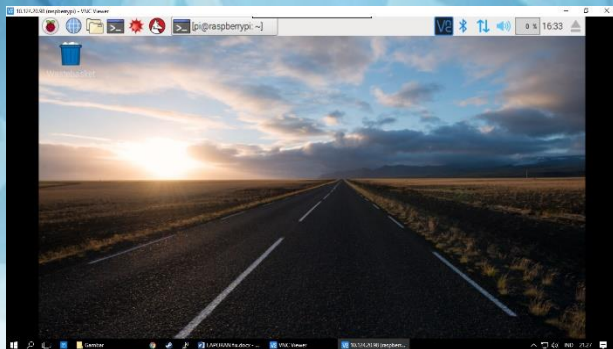


3. Laptop



Cara peng

1. Pastikan raspberry pi sudah tersambung pada plant purifikasi biogas
2. Nyalakan raspberry pi dan akan tampil layar pada gambar berikut



3. Akan keluar tampilan seperti ini apabila data sudah masuk ke database

TA BIOGAS



00:00:00

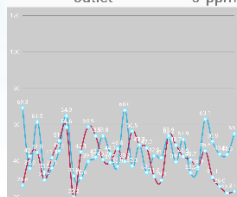
00:00:00

Saat ini inlet

0 ppm

outlet

0 ppm



PROSEDUR PENGGUNAAN

Untuk mengaktifkan plant purifikasi biogas ini perlu diperhatikan cara operasionalnya, yaitu sebagai berikut:

1. Pastikan semua wiring rangkaian sudah terpasang dengan benar dan baik.
2. Pastikan tidak ada kebocoran pada masing-masing bagian plant
3. Pastikan pemasangan sensor sudah dilakukan dengan baik dan benar.
4. Pastikan sambungan kabel yang terhubung dengan tegangan AC terhubung dengan benar dan pastikan tidak ada kabel yang terkelupas.
5. Pastikan apakah power supply untuk controller dan bagian-bagian lainnya telah terpasang dan terhubung dengan benar.
6. Sambungkan selang pada inlet purifikasi. Selang harus berukuran 5x8 mm agar dapat masuk ke sambungan pipa inlet purifikasi
7. pasang kabel pada stopkontak.
8. Nyalakan MCB yang berada pada dalam panel untuk menyalakan power supply 12V.
9. Tekan tombol ON/OFF pada pintu panel box yang berfungsi untuk menyalakan power supply 5V.
10. Maka sistem purifikasi berhasil dinyalakan dan siap digunakan.

BIODATA PENULIS



Penulis dilahirkan di Sidoarjo pada tanggal 15 Februari 1995. Pada saat ini penulis bertempat tinggal di Ds. Karangpuri Dsn. Sampuri RT 01 RW 03, Wonoayu, Sidorajo, Jawa Timur. Pada tahun 2007, penulis telah menyelesaikan pendidikan dasar selama 6 tahun di SDN Karangpuri 1 Wonoayu. Pada tahun 2010, penulis telah menyelesaikan pendidikan tingkat menengah pertama di MTsN Krian Sidoarjo selama 3 tahun. Pada tahun 2014, penulis telah menyelesaikan pendidikan tingkat menengah atas di MAN Sidoarjo selama 3 tahun. Pada tahun 2017 ini, penulis mampu menyelesaikan pendidikan Diploma di Program Studi D3 Teknik Indrumentasi, Departemen Teknik Indrumentasi, Fakultas Teknologi Vokasi, Institut Teknologi Sepuluh Nopember Surabaya. Penulis dapat dihubungi melalui alamat email indraradian@gmail.com.